AD050081

**DEFENSE COMMUNICATIONS AGENCY**

COMMAND AND CONTROL
TECHNICAL CENTER
WASHINGTON, D C. 20301

(2)

B S.

AD A094536

LEVEL II

IN REPLY
REFER TO C314

(11) 30 June 1979

(12) 72

TO: RECIPIENTS

(14) CCTC-

SUBJECT: Change 2 to Program Maintenance Manual CSM-MM-9-77-V2-Chg-2
Volume II, Weapon/Target Identification Subsystem

1. Insert the enclosed change pages including page 181 which was printed in error. Destroy the replaced pages according to applicable security regulations.

2. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page.

3. When this change has been posted, make an entry in the Record of Changes.

FOR THE DIRECTOR

J. DOUGLAS POTTER
Assistant to the Director
for Administration

71 Enclosures
Change 2 pages

(6) The CCTC Quick-Reacting General War Gaming System (Quick) Program Maintenance Manual. Volume II. Weapon/Target Identification Subsystem. Change 2.

DTIC
ELECTE
FEB 2 1981
S D
C

DDC FILE COPY

**DISTRIBUTION STATEMENT A**
Approved for public release;
Distribution Unlimited

81 1 19 222

409658

AMERICAN REVOLUTION BICENTENNIAL 1776 1976

## EFFECTIVE PAGES - MAY 1979

This list is used to verify the accuracy of CSM MM 9-77, Volume II after change 2 pages have been inserted. Original pages are indicated by the letter 0, change 1 pages by the numeral 1, and change 2 pages by the numeral 2.

| Page No. | Change No. |
|---|---|
| Title Page | 0 |
| ii | 2 |
| iii | 0 |
| iv | 2 |
| v-vi | 0 |
| 1-2 | 0 |
| 3 | 1 |
| 4 | 0 |
| 5 | 1 |
| 6 | 0 |
| 7 | 2 |
| 8-9 | 0 |
| 10 | 2 |
| 10.1-10.2 | 1 |
| 11-46 | 0 |
| 47 | 1 |
| 48-49 | 0 |
| 50 | 1 |
| 51 | 0 |
| 52-53 | 1 |
| 54 | 0 |
| 55-56 | 1 |
| 57-58 | 0 |
| 59-63 | 1 |
| 64-76 | 0 |
| 77 | 2 |
| 78-79 | 0 |
| 80 | 2 |
| 81 | 1 |
| 82-83 | 0 |
| 84-86 | 1 |
| 87-89 | 0 |
| 90 | 1 |
| 91-96 | 0 |
| 97 | 2 |
| 98 | 0 |
| 99-100 | 2 |
| 101-104 | 0 |
| 105-106 | 1 |
| 107-109 | 0 |
| 110-114 | 2 |
| 114.1-114.2 | 2 |
| 115-117 | 2 |
| 118-120 | 0 |
| 121-123 | 1 |
| 124-129 | 0 |
| 130-132 | 2 |
| 133 | 0 |
| 134-135 | 2 |
| 136-138 | 0 |
| 139-140 | 2 |
| 141 | 0 |
| 142-147 | 2 |
| 148 | 0 |
| 149-152 | 2 |
| 153 | 0 |
| 154 | 2 |
| 155-156 | 0 |
| 157-167 | 2 |
| 167.1-167.2 | 2 |
| 168-172 | 2 |
| 173-178 | 0 |
| 179-202 | 1 |

ACKNOWLEDGMENT

This document was prepared under the direction of the Chief for Military Studies and Analyses, CCTC, in response to a requirement of the Studies, Analysis, and Gaming Agency, Organization of the Joint Chiefs of Staff. Technical support was provided by System Sciences, Incorporated under Contract Number DCA100-75-C-0019. Change set two was prepared under Contract Number DCA100-78-0035.

| Section | | Page |
|---|---|---|

| COLS. | ITEM | USED# | CREATED BY JLM |
|---|---|---|---|
| 205 | Scaling Factor | | |
| 206-208 | Radius | JRADIUS | |
| 209-212 | Percent Capacity | | |
| 213-224 | Dimensions | | |
| 225-236 | Fiscal Year Projections | | |
| 237 | File ID Code | | |
| 238 | Phase Code | | |
| 239-245 | Security Class | | |
| 246-247 | Remark | | |
| 248-253 | Owner UIC | | |
| 254-255 | Serv Spcl Code | | |
| 256-258 | READY Code | | |
| 259-267 | Blank | | |
| 268-283 | Not Used | | |
| 284-286 | SIOP Table Number | | |
| 287-288 | Blank | | |
| 289-293 | DESIG | | DESIG |
| 294 | Flag if in the Data Base | | * or blank |
| 295-300 | Type | | TYPE |
| 301-303 | Not Used | | |
| 304-306 | Subset of Class Index | | |
| 307-318 | Not Used | | |
| 319 | QUICK Region | | IREG |
| 320 | SAGA Region | | * |
| 321-324 | Not Used | | |
| 325-330 | SAGA Flag | | |
| 331-336 | BLANK | | |

#Variables are named as used in common block JADREC

Figure 3. (Part 2 of 2)

An optional output of JLM is a JAD format file from the selected targets for use by damage assessment systems external to QUICK. The format is as shown in figure 3 with the added entries created by JLM.

## 2.4 Concept of Operation

The function of the JLM is to build portions (targets) of the integrated data base by selecting records from a file that is in a JAD format. JLM operates in three modes. First, a section of the integrated data base called the Assignment Table is built through user inputs. This table describes what sort of target is to be added to the data base and how it will be included. Second, given a completed Assignment Table, the selection of JAD records is executed and a Damage Assessment tape is prepared for use in processors external to the QUICK system. Third, after record selection, provisions are included for deleting individual records not required for QUICK processing. In the text English sense verbs ASSIGN, SELECT, and ASTERISK initiate the three JLM functions.

## 2.5 Identification of Subroutine Functions

2.5.1 Subroutine ASSIGN. This subroutine is the first subroutine within overlay link ASSI executed upon the appearance of verb ASSIGN. Depending upon the adverb; subroutines ALPHAS and/or PLAYERS are executed.

2.5.1.1 Subroutine ALPHAS. The ALPHAS clause builds the bulk of the Assignment Table and is performed by this subroutine. The main portion of subroutine ALPHAS involves the reading and correct definition of the generalized input clause. This clause is fully detailed within section 2 of Users Manual, Volume II. Major points are repeated here for purposes of outlining the major thrust of the code written for subroutine ALPHAS. The input clause has the form:

ASSIGN ALPHAS side // class - type ≥ [ {minimum-capacity | name} ]

/ low-catcode [- high-catcode] [* task] ,

desig-alphabetic [, alternate-desig . . .]

[ [NOT] {OWNED BY | LOCATED IN} country-code [, country-code . . .] ]

General comments are:

- o The side must be first
- o The target class must be preceded by two slashes
- o The target type must be preceded by a dash
- o If minimum capacity or name is used, it must be preceded by a greater than symbol
- o The lowest catcode must be preceded by one slash and if a range of catcodes are used the highest catcode is preceded by a dash
- o TASK is preceded by an asterisk and DESIG by a comma
- o Country codes are preceded by either OWNED or IN if the assignment is restricted

2.5.1.2 Subroutine PLAYERS. Similar to ALPHAS, subroutine PLAYERS reads the input clause and updates the Assignment Table. The generalized input command is:

ASSIGN PLAYERS side // region / country-code

[, country-code . . .] [side // region . . .]

The side must be first, the region must be preceded by two slashes and the list of country codes must be preceded by a single slash.

2.5.2 Subroutine SELECT. The use of the SELECT verb (and hence the execution of overlay link (SELE)) instructs the JLM to select records from the JAD format input file according to the developed Assignment Table. The SELECT command has a maximum of six optional adverbs and are:

- o WHERE - normal WHERE clause without OF or LIKE
- o UNIT - used to define input unit if it is not 20
- o ONPRINTS - causes the print of the output JAD format
- o REPLACING or OMITTING - used to replace existing targets or to ignore duplicates
- o ORDER - allows the user to specify the arrangement that the classes will be added to the integrated data base
- o SETTING - used to set the value of TARDEF to allow for automatic assignment of values for attributes TARDEFHI and TARDEFLO

If the user desires to bypass the Assignment Table, the expanded functions of the adverbs UNIT and ORDER may be used and is indicated by adding 100 to the logical unit number in the UNIT clause. The ORDER clause now takes on additional meaning. The order of the target class names in the ORDER clause must correspond with the set number on the input file. Also the last two entries must be the SIDE (RED or BLUE) followed by the type of file being input, i.e., BTL, BTB, DBASSES.

If the Bypass option is used, all data on the input file is directly added to the data base using subroutine ADTOBASE. No preprocessing is done.

After user input definition, SELECT reads a JAD input record and queries the Assignment table (subroutine FNDTAR) to ascertain if the target should be added to QUICKs data base. In it is not to be added, the next JAD input record is read. Otherwise investigations are made for exclusion of the input record due to a WHERE clause. For each selected target record, data is written onto files 25 and 21.

After all target records have been selected, data files are sorted and read. For each record, subroutine ADTOBASE is called for the definition of the selected target record onto QUICKs data base.

Finally tests and code are made to quarantee the JAD records are in proper sort and additionally JAD selected records are printed if user directed.

2.5.2.1 Subroutine ADTOBASE. This subroutine adds the data from the selected JAD record onto QUICKs data base. In addition to inserting attribute values, this subroutine places the target record on the proper chains. This includes linkage under the vulnerability, class, type, and other headings as directed by the nature of the target.

2.5.3 Subroutine ASTERISK. This overlay removes targets from the integrated data base and flags all target records on an output JAD format file. If identical target records reside both within the integrated data base and an output JAD file, an asterisk (character position 294) is placed on that record within the JAD file.

The list of target record to be retained are defined within a KEEPING clause and has the form:

KEEPING lowdesig [- highdesig]

[, lowdesig [- highdesig] . . .]]

This clause consists of a list of DESIG ranges that are to be kept in the data base and flagged on the output file.

SECTION 3. DBMOD MODULE

3.1 Purpose

The purpose of DBMOD is to alter the content or characteristics of a data base in order to adapt the data base to the specific scenerio for which the plan is being developed. Because of its highly specialized nature, module DBMOD should be examined for possible revision each time a new plan is to be generated.

3.2 Input

User commands plus the integrated data base are necessary inputs to DBMOD. User inputs define the scenario, attacking and defending sides, plus optional inputs whereby nondefault scaling factors may be set.

All targets to be processed by the QUICK system must have been defined prior to DBMOD execution. This also includes a definition for each target's value (attributes VAL, IGIW or POP). For the attacking side attributes ADBLI, NADBLI, or ADBLR and NADBLR, and NPRSQ1, NPRSQ2, or NPRSQ3 must also have been defined.

3.3 Output

DBMOD generates printed reports and modifies the integrated data base for all class targets treated as U/I for the defending side and for all missile and bomber class targets for the attack side. U/I type targets modify attribute VAL. Missile and bomber class targets modify attributes NOINCO, NALERT, NOPERSQ, ALRTDB and NLRTDB. If user requested, attributes TARDEFHI and TARDEFLO are modified.

3.4 Concept of Operation

DBMOD begins by reading input user commands and stores values that define the scenerio to be constructed, the attacking and defending sides, U/I class names, and the nondefault scaling factors used for U/I class value calculations. DBMOD, then determines the attributes for NOINCO (number in commission) and NALERT (number on alert) for bombers and missiles. The user also has the option of scaling the value (VAL) given to an U/I target based on the values for population (POP) and IGIW. The option also exists to calculate local bomber defenses (attributes TARDEFHI and TARDEFLO). For given collections of targets records, parameters are summed and properly printed.

3.5 Identification of Subroutine Function

3.5.1 Subroutine DESTAB. With the exception of utility routines, DESTAB is the only subroutine included under DBMOD. DESTAB keeps track of the number of target records within the data base, the number of target records deleted, and produces summary prints.

## 3.6 <u>Common Block Definition</u>

Common blocks used by DBMOD are outlined in table 2. Common blocks that communicate with the COP are given in appendix A of Maintenance Manual, Volume I.

Table 2. Module DBMOD Internal Common Blocks

| BLOCK | VARIABLE OR ARRAY | DESCRIPTION |
|---|---|---|
| CLASSES | CLASSES(60) | List of all legal class values from FINDCLAS |
| LOCAL | LOCAL(1) | Attacking side |
| | LOCAL(2) | Defending side |
| | SCENARIO | SIERRA, INDIA, or ROMEO input |
| | AUTARD | Flag to automaticaly generate TARDEFHI and TARDEFLO |
| | PCTTIW, PFIW, PCTPOP, PFPOP | Constants used in U/I scaling equations. Default values are: PCTIW=3.06, PFIW=.81, PFPOP=PCTPOP=0 |
| SIDES | SIDES(5) | Common with FINDSIDE and stores all sides with headers in the data base. |
| TARDEF | NTARHI | Level of local bomber defense at high altitude |
| | NTARLO | Level of local bomber defense at low altitude |

### 3.7 Subroutine ENTMOD

| | |
|---|---|
| PURPOSE: | Alter data base to specified scenario |
| ENTRY POINTS: | ENTMOD (first subroutine called when overlay link DBMOD is executed) |
| FORMAL PARAMETERS: | None |
| COMMON BLOCKS: | CLASSES, C10, C15, OOPS, LOCAL, SIDES, TARDEF |
| SUBROUTINES CALLED: | DIRECT, DROPDES, FINDCLAS, FINDSIDE, HDFND, HEADRF, INSGET, KEEPDES, MODFY, NEXTTT, PRNTDES, RETRV, SETDEF |

Method:

DBMOD initially reads user's inputs and stores needed parameters. The verb for this module is MODIFY and adverbs are UICLASSES and SETTING. Within the SETTING clause parameters SCENARIO, ASIDE and DSIDE must be defined. SCENARIO defines how the data base is shaped and recognizes values of SIERRA, INDIA, or ROMEO. Parameter ASIDE function is to set the attacking side and DSIDE to set the defending side. Sides are set within record 'NUMTBL' (see figure 15). The UICLASSES clauses permits a series of target classes to be entered which informs DBMOD to test as if a U/I target.

If desired the user may request calculations for local bomber defenses through an input within the clause of setting TARDEF=YES. The remaining allowable inputs to the clause are the setting of variables PCTIW, PFIW, PCTPOP, or PFPOP to non-default values for use in U/I calculations.

After input definition, the individual tasks performed are:

a. The appropriate number of bombers or tankers for each squadron (NOPERSQN) is selected depending upon the particular plan being developed (Initiative, Surprise, or Retaliatory)

b. The number of bombers or tankers in commission (NOINCOM) for each squadron is calculated by specifying that attribute NOINCOM is equal to NOPERSQN

c. The number of bombers or tankers which are on alert (NALERT) for each squadron is also set to NOPERSQN

d. The relative value (attribute VAL) of urban/industrial targets is calculated as a function of general industrial worth (IGIW) and population (POP). As a default condition VAL is calculated for each individual target under target classes that equals U/I or IGIW. In addition, VAL is calculated similarly for each target class supplied within the user input UICLASSES clause.

SECTION 4. INDEXER MODULE

## 4.1 Purpose

To provide for economical handling of data and to facilitate communications between QUICK modules, it is necessary to assign indices to various data contained in the data base. Module INDEXER is designed to perform this task. In addition, INDEXER processes all potential targets and, where appropriate, forms them into complex targets.

## 4.2 Input

The input to module INDEXER consists of the user input command and the integrated data base. The user-input command identifies a complexing option or a weapon yield used for complexing and print option requests.

INDEXER accesses the integrated data base in two ways. First, it accesses the target data by retrieving the target headers (TGTHD), the target type records (TARGTY) on the TGTTYP chain, the TARGET records on the TGTTGT chain, and finally the MSBMTG records on the TARGXX chain. Second, it accesses targets on the CMPTGT chain which will contain all targets stored.

## 4.3 Output

INDEXER modifies the integrated data base in two ways. First, it alters the contents of the TARGET record, adding an index number and, for missile and bomber targets, calculates the time decay values. Also, as a result of the complex formation process, it changes the linkage of targets on the CMPTGT chain. When originally stored, all TARGET records are placed on the CMPTGT chain as details of a single master: This master record (type COMPTG) may be thought of as the master of a chain of all simple targets. Following the complex formation process, each complex causes the creation of a COMPTG record. This record is used as the master of a CMPTGT chain whose details are those TARGET records which make up the complex. After the creation of this record, all TARGET records which are a part of this complex are modified so that they are now chained to the new record rather than the simple target master. Thus when all processing is completed, all targets will be differentiated as to whether they are simple, or complex targets by the COMPTG record to which they are chained.

## 4.4 Concept of Operation

After a scenario has been selected, module INDEXER performs necessary calculations and additions to the refined data base. The major objectives of INDEXER are to: (a) assign unique indices to all targetable records (referred to as index number, attribute INDEXNO); (b) automatically calculate time decaying value points for all target bomber and missile bases; (c) calculate for each unique target vulnerability a complexing lethal radius based on user selected yields; (d) complex individual targets based on selected algorithm; and (e) define the target complex classes.

## 4.5 Identification of Subroutine Functions

4.5.1 Subroutine COMPLEX. This subroutine queries all potential targets on an earth sector (boundaries of longitude) basis and forms complexes. Simply, elements of targets are defined as being in a complex if they are geographically within a defined destruct radius of each other. The destruct radius is the lethal radius calculated and stored in earlier processing.

4.5.2 Subroutine CRTBLE. This subroutine calculates complexing lethal radius based on hard coded tables and is executed only when user directed.

4.5.3 Subroutine SETVAL. If a target belongs to a missile or bomber class and is salvoed, this routine will calculate time value decay curves based on the rate at which sorties leave a launch base.

## 4.6 Common Block Definition

Common blocks used by INDEXER are outlined in table 3. Common blocks that communicate with the COP are given in appendix A of Program Maintenance Manual, Volume I.

Table 3. Module INDEXER Internal Common Blocks

| BLOCK | VARIABLE OR ARRAY | DESCRIPTION |
|---|---|---|
| CYIELD | SYIELD(2) | Yield, in megatons, used in forming complexes. Value is user determined. |
| DIFFLAT | DIFFLAT | Maximum difference in latitude (DEG) in forming complexes. Computed in INDEXER based on the softest target in the data base. |
| IOPRT | IOPRT | If zero, nonstandard prints are suppressed. User determined. |
| KEEP | | As used within ENTMOD, contains local controling parameters (used to reduce core space). As used within subroutine COMPLEX, contains varying definitions of target data. |
| NMCLAS | | Contains the class name for the NUMTBL record. |

## 4.7 Subroutine ENTMOD

| | |
|---|---|
| PURPOSE: | Read user inputs, calculate and store complexing lethal radius, determine attribute INDEXNO and control flow of supporting subroutines |
| ENTRY POINTS: | ENTMOD (first subroutine called when overlay link INDXER is executed) |
| FORMAL PARAMETERS: | None |
| COMMON BLOCKS: | C10, C15, C25, C20, C30, CYIELD, DIFFLAT, IOPRT, KEEP |
| SUBROUTINES CALLED: | COMPLEX, CRTBLE, DIRECT, HDFND, HEAD, INSGET, ITLE, MODFY, NEXTTT, RETRV, SETVAL, VLRADI |
| CALLED BY: | COP |

Method:

Module INDEXER begins (figure 17) by reading (through utility subroutine INSGET) and storing user input parameters, then for each unique vulnerability contained within the data base, a complexing lethal radius is calculated and stored. Following this, individual targets are chained in a specified manner and modified to include attribute INDEXNO. As individual targets are chained, subroutine SETVAL is called for all missile and bomber classes for possible time value decay calculations. After querying targets, subroutine COMPLEX is called in order to form target complexes and upon completion processing is terminated.

User Input Definition

INDEXER initially retrieves record type 'NUMTBL' in order to define the attacking (ASIDE) and defending (DSIDE) side. These attributes were stored in the data base by module DBMOD. Following side definition, the user inputs are retrieved and needed values stored.

If the verb is correct (comparison to local parameter IND) processing continues; otherwise an error message is printed and processing stops.

Existence of adverbs WITH, VNOPTION, or ONPRINTS are checked for. Use of adverb ONPRINTS, implies nonstandard prints are to be produced; adverb VNOPTION implies that complexing lethal radius is to be obtained from hard coded tables. In absence of the VNOPTION clause, complexing is performed with an assumed weapon yield of one megaton. The user may override this yield through a clause introduced by the adverb WITH. Both attributes YIELD and SIDE are included within the WITH clause.

may exist more than one 'TGTTYP' record. This is possible since the attributes defined in the 'TGTTYP' records may have different values for a TYPE value. For instance, for TYPE=MMIII there could be two entries for attribute CNTRYL (country location), say US and CA. For this condition two 'TGTTYP' records will exist and both have attribute TYPE=MMIII. Therefore upon chaining 'TGTTYP', the entire list must be checked for multiple occurrences of the same TYPE value.

After the last class entry for side 1 is processed, checks are made to ensure all side 2 entries have been processed. This is necessary since the major processing is for all side 1 entries and the fact exists that class names may be defined for side 2 but not side 1. Local array ICHK is set to nonzero as each side 2 class name is processed.

## 4.8 Subroutine COMPLEX

| | |
|---|---|
| PURPOSE: | To form complex targets |
| ENTRY POINTS: | COMPLEX |
| FORMAL PARAMETERS: | None |
| COMMON BLOCKS: | C10, C15, C30, DIFFLAT, IOPRT, KEEP, NMCLAS |
| SUBROUTINES CALLED: | DIRECT, DLETE, HDFND, KEYMAKE, GLOG, MODFY, NEXTTT, ORDER, RETRV, SLOG, STORE, TIMEME |
| CALLED BY: | ENTMOD (of overlay link INDEXER) |

Method:

Individual target records are queried in order to form target complexes. If any two targets are geographically located within one half the sum of the complexing lethal radius of each target, they belong to the same complex. For each new complex formed, a complex number (parameter ICOMPL) is sequentially updated, stored under record 'COMPTG' and each individual target belonging to the complex is stored on the 'CMPTGT' chain.

Each target stored within the data base is retrieved, sorted on increasing latitude (methodology outlined in next subsection), and final results written onto data file unit number IR. Data file IR is then read, target data stored in working arrays /KEEP/, and complexes formed.

Complexing Algorithm

The search for complex targets begins by comparing differences in latitude for consecutive targets sorted by increasing latitude, beginning with the first noncomplex target. When a distance between the first selected target (associated with latitude CLATI) and any other individual (latitude CLATJ) that has not as yet been complexed is less than one half the sum of the lethal radius of each target, the target associated with latitude CLATJ is said to belong to a complex associated with latitude CLATI. Array LCOMP is updated to record this occurrence.

Target CLATI continues to be tested against subsequent targets in the sorted file until a difference in latitude greater than parameter DIFFLAT is encountered. DIFFLAT is the maximum complexing lethal radius defined within the game. Targets included in the list LCOMP are now compared in the same way to find additional members of the complex. The process is repeated until all targets in the list LCOMP have been investigated, and the complex is completed. The complex then is assigned the next value of ICOMPL and each member within the complex are properly chained.

## Target Latitude Sort

Data based stored targets are sorted on increasing latitude in order to simplify the complexing burden. This reordering is performed by a basic sort/merge technique and is as follows:

- o Retrieve a data base target record
- o Store target latitude in array KEY1 and target reference code in array KEY3.
- o When working arrays are filled branch to label 9000 to sort the arrays and merge with any previous sorts. Arrays are sorted via the standard QUICK subroutine ORDER. If there were multiple calls to label 9000, results were written onto file unit IR. This file is read and results merged with contents in working arrays.
- o After sort/merge has been completed, flip data read and write files (IR and IW) reset pointers to initial conditions and continue IDS query.

After this process is complete data unit IR contains all target records in latitude sort. Each record contains two words: target latitude and target reference code.

## Target Data Retrieval

After targets are sorted, complexes are formed by querying data file IR and retrieving additional data base information. The size of QUICK's data base mandates that IDS interface be minimal. Therefore needed target data in stored in working arrays KEY1, KEY2, KEY3. Latitude and the fraction portion of lethal radius is packed in KEY1; Longitude, complex flag indicator, and the integer portion of the lethal radius is packed in KEY2; and the reference code is stored in KEY3. Up to 3,800 individual targets may be contained in the working arrays.

As complexes are formed information is obtained from the working arrays as directed by the controlling pointer INDEX. When INDEX exceeds array limits, complexing must temporarily halt and working arrays redefined. At this juncture, processing has been completed for all records starting with the first entry into the working arrays up to location LCOMP(1) minus one (recall that LCOMP contains the indexes of potential complexes). With this knowledge, working arrays are compressed, file IR read, and working arrays filled with additional target data. This process begins at label 565.

Code must honor the condition ·  ·e arrays can no longer be compressed (LCOMP(1)=1). If this occurs, target data is written onto an indexed random file (ISAMLUN) and query, then, is from that file.

Subroutine COMPLEX is illustrated in figure 18.

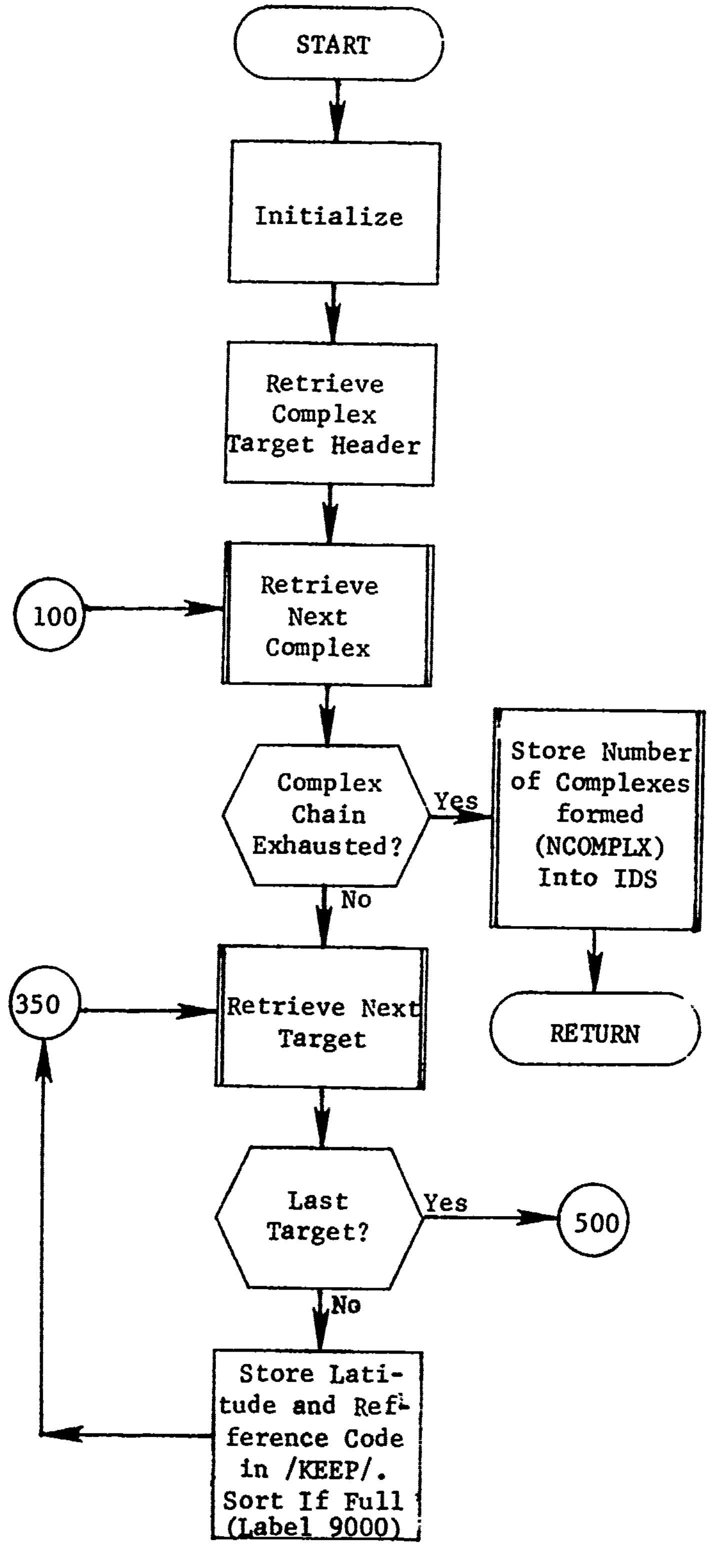


Figure 18. Subroutine COMPLEX (Part 1 of 8)

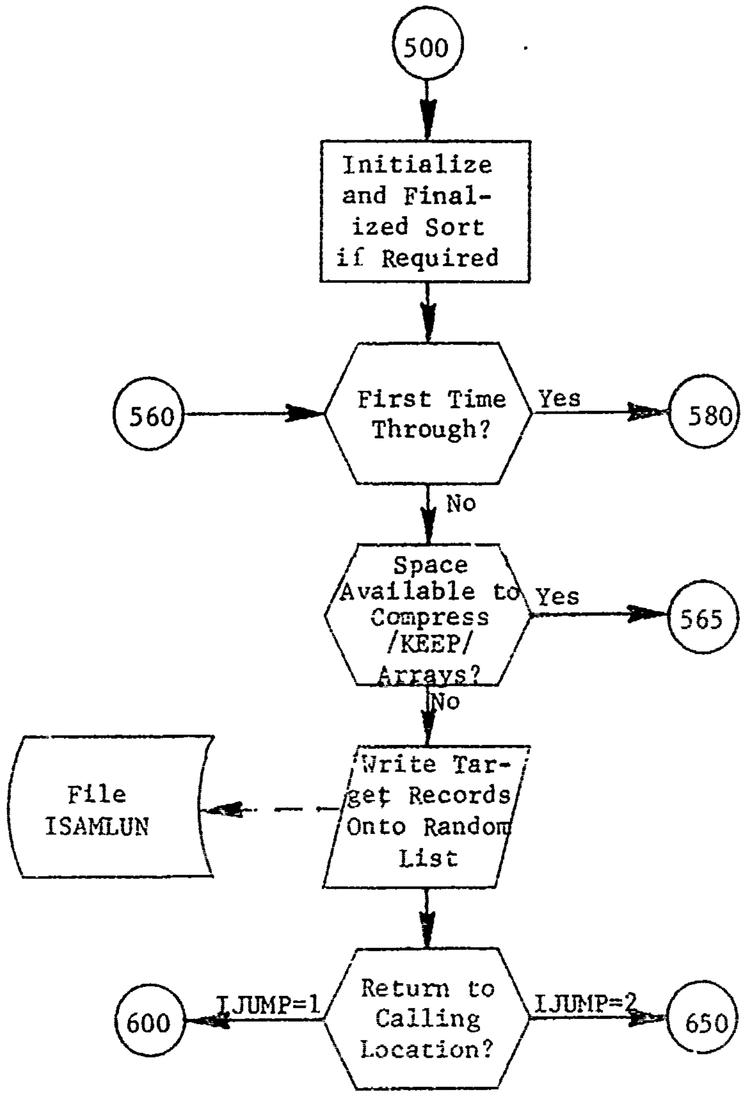


Figure 18. (Part 2 of 8)

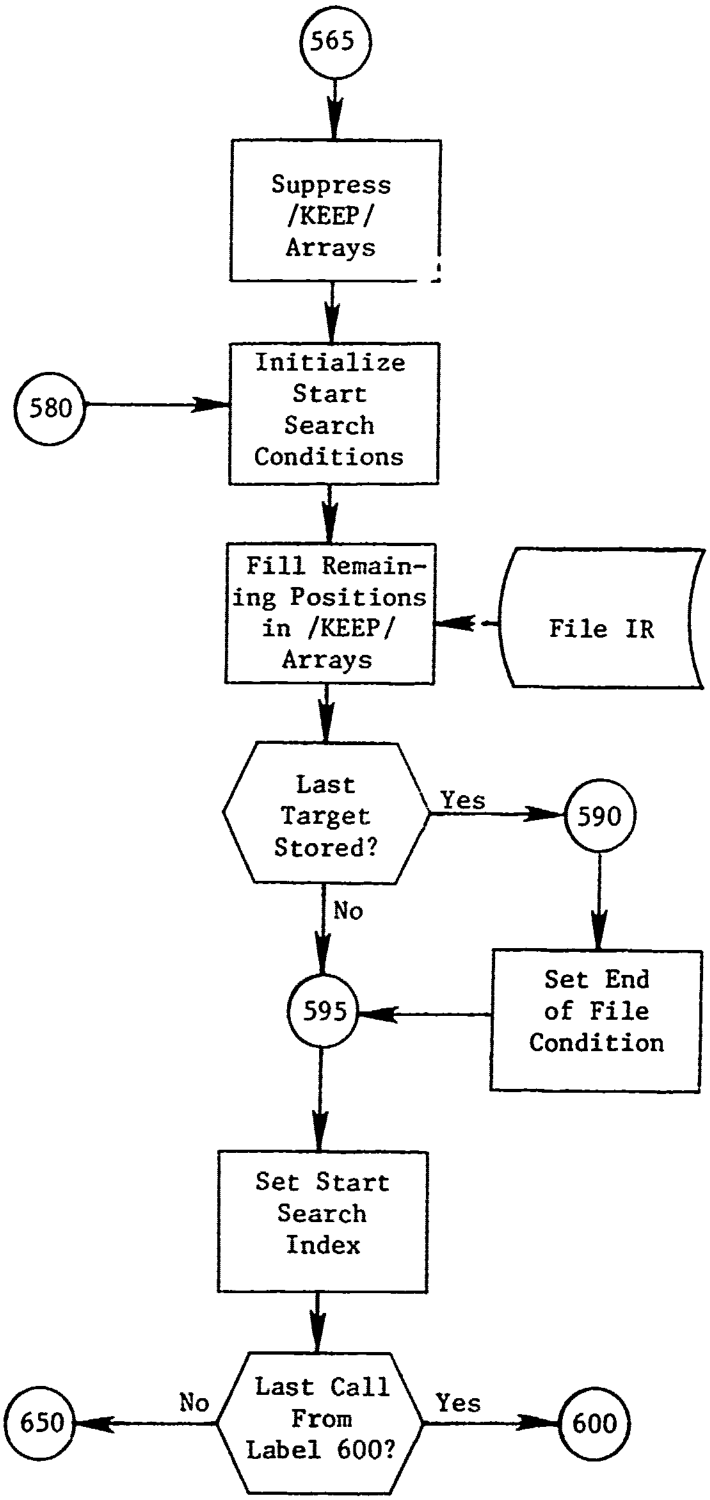


Figure 18. (Part 3 of 8)

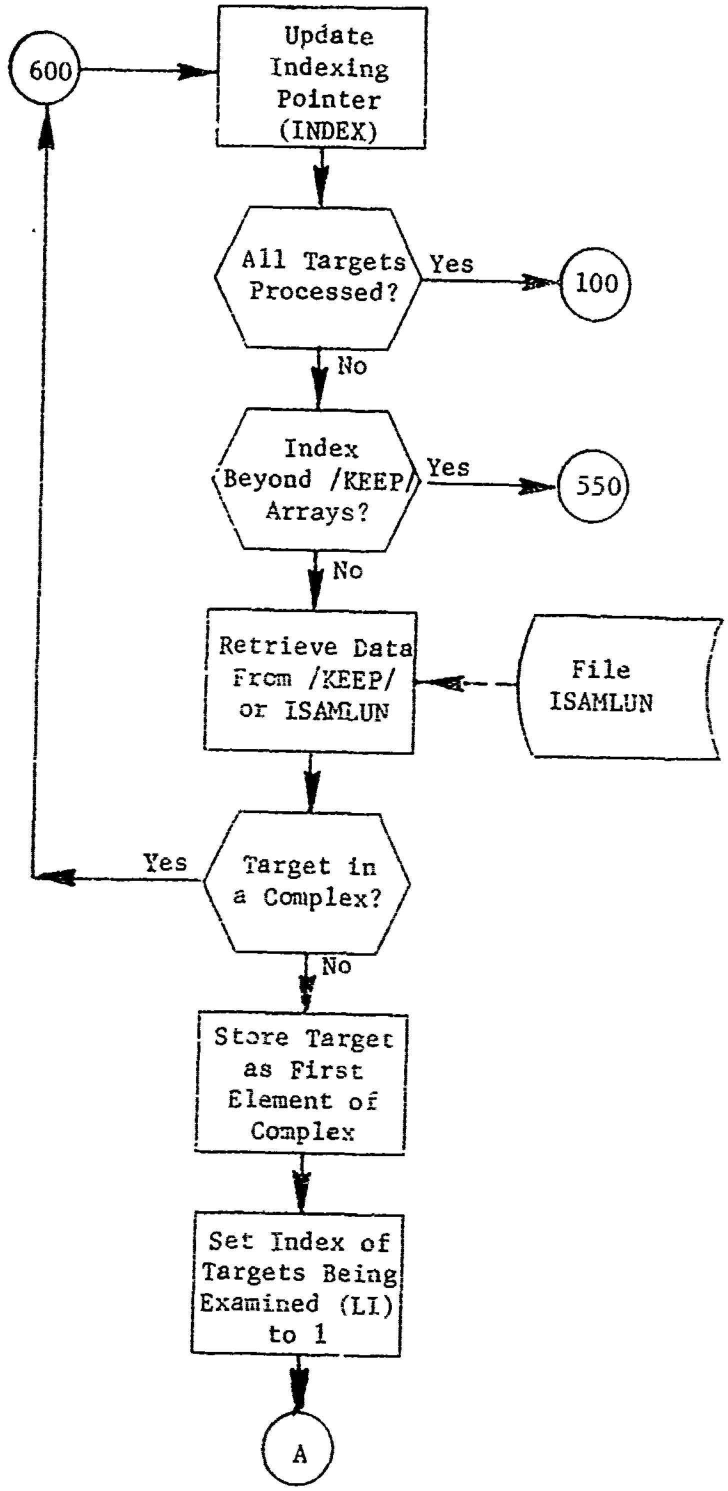


Figure 18. (Part 4 of 8)

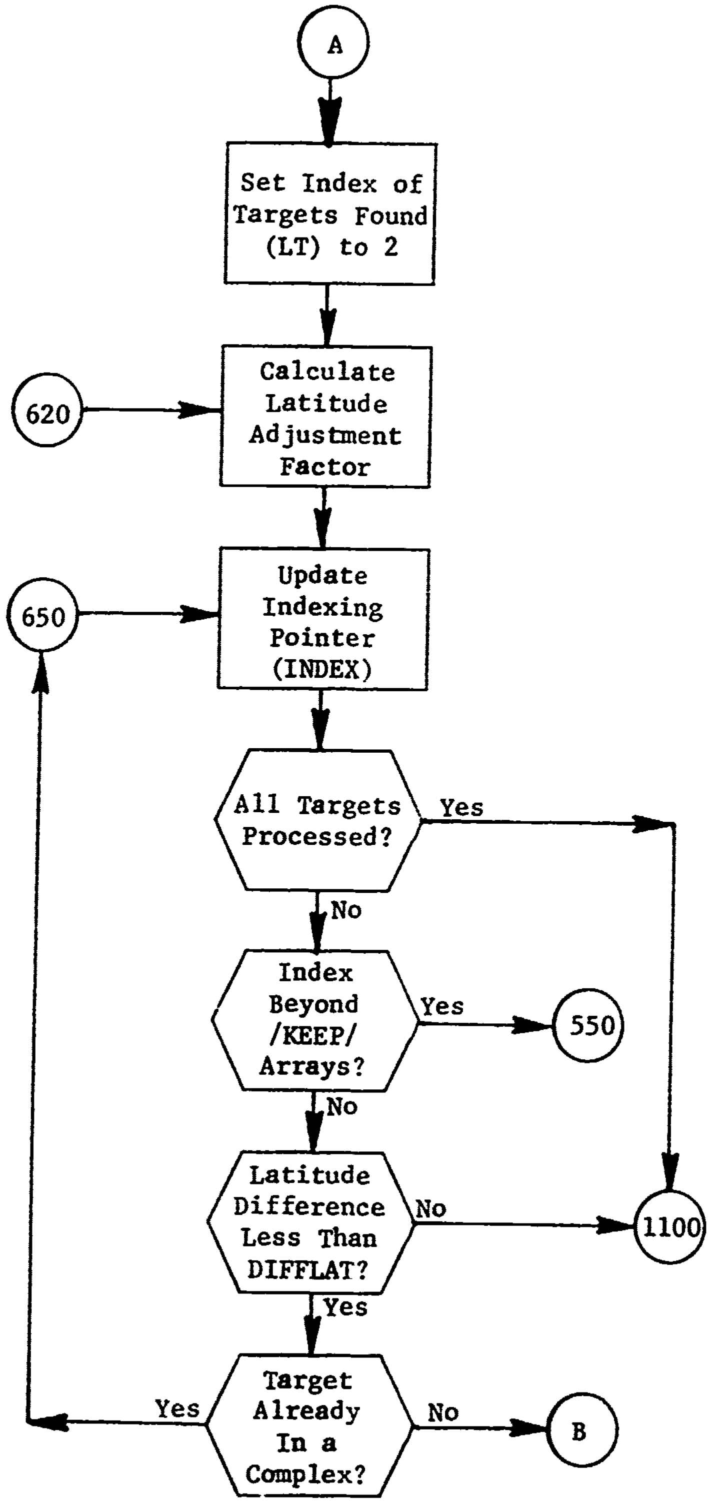


Figure 18. (Part 5 of 8)

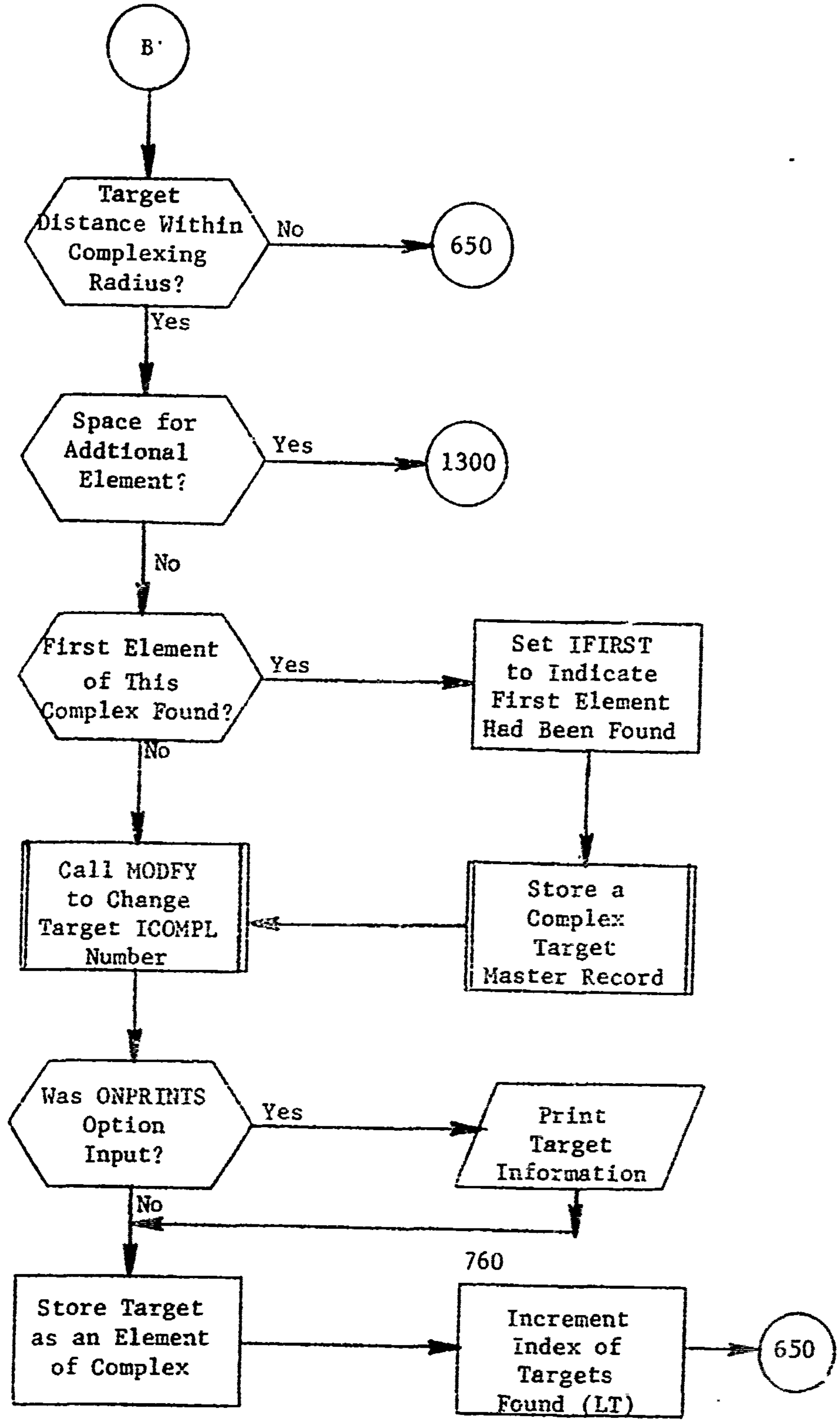


Figure 18. (Part 6 of 8)

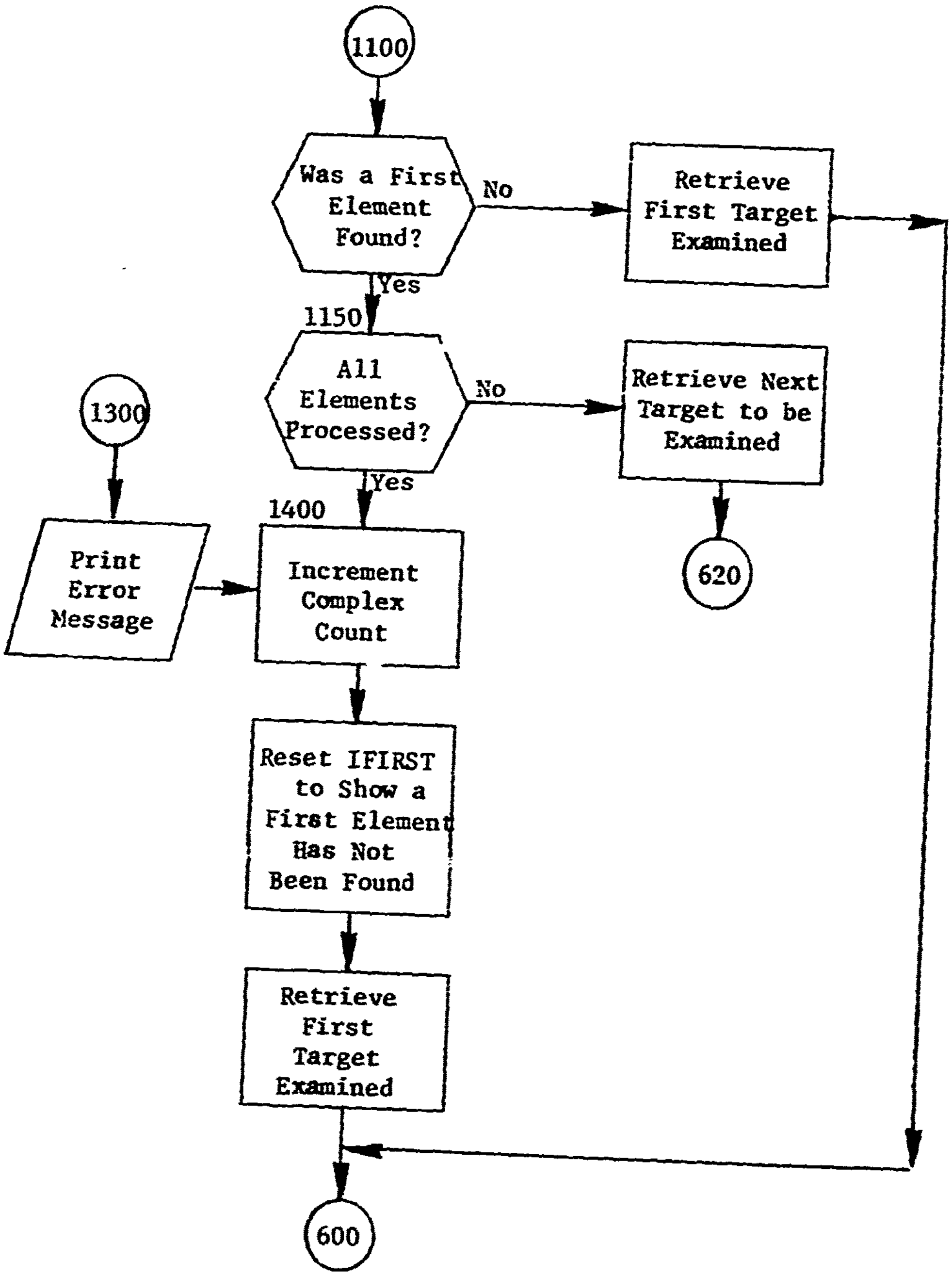


Figure 18. (Part 7 of 8)

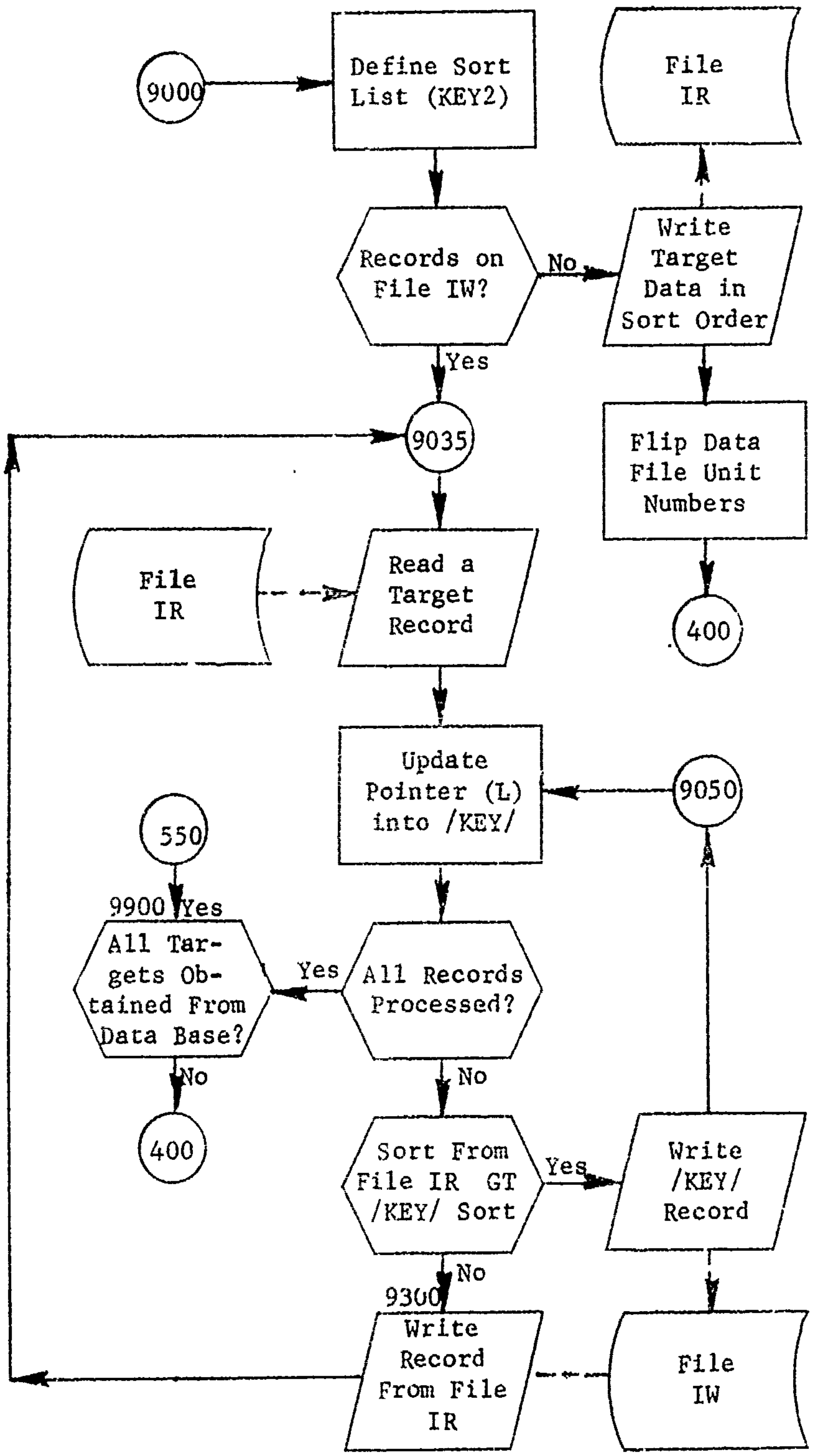


Figure 18. (Part 8 of 8)

## 4.9 Function CRTBLE

PURPOSE: To calculate complexing lethal radius

ENTRY POINTS: CRTBLE

FORMAL PARAMETERS: IVN, vulnerability

COMMON BLOCKS: None

SUBROUTINES CALLED: VLRADI

CALLED BY: ENTMOD (of overlay link INDXER)

Method:

Based on input parameter (IVN) complexing lethal radius is obtained by proper indexing into hard coded arrays (Q and P) which is the letter portion of vulnerabilities. Defining a yield as equaling the first two integers (in megatons) of IVN, subroutine VLRADI is called to find an adjusted VN which is turn is used as the index into the P and Q arrays.

Function CRTBLE is illustrated in figure 19.

compute the final normalized class value factors. Also, the summation of the VALs for each record is so scaled such that its result equals 1000. This scaling allows for allocation evaluations between various data bases.

Multiple targets are made up for missile sites which do not belong to a complex. A multiple target consists of at least two and not more than five consecutively indexed sites from the same squadron.

## 5.6 Internal Common Blocks

All of the common blocks used by module PLANSET are given in table 4. Common blocks which communicate with the COP are given in appendix A of Maintenance Manual, Volume I.

Table 4. Module PLANSET Internal Common Blocks (Part 1 of 2)

| BLOCK | VARIABLE OR ARRAY | DESCRIPTION |
|---|---|---|
| CPRIOR | MAXDSG | Maximum number of input DESIGs alpha portions. Used for representative target definition |
| | MAXTASK | Maximum number of input TASKs. Used for representative target definition |
| | ITSK | Number of entries in array IPTSK |
| | IDSG | Number of entries in array IPDSG |
| | IPTSK(48) | Stores TASK inputs for defining representative target of complexes |
| | IPDSG(200) | Stores DESIG alpha portion input for defining representative target of complexes |
| | ISUBT | Flag indicating whether TASK inputs are 1 or 2 characters long |
| EXCLAS | UCLASS(15) | Stores names of target classes for defending side |
| | UCREF(15) | Stores reference codes for target class headers for defending side |
| | VALFAC(15) | Stores scaling factors for each target class |
| INZ | I1, I2, I3 | Local storage for INSGET's processing |
| KEEP | | Local storage whose meaning varied during the computation process. |
| MASK | MASK1 | Testing parameter for attribute TASK |
| | MASK2 | Testing parameter for attribute TASK |
| PBLOK | PBLOK(31,10) | Used in SRTTGT for target designator, number print and in PRINTGP for weapon and group prints |
| | ARRAY(750) | Temp storage |

Table 4. (Part 2 of 2)

| BLOCK | VARIABLE OR ARRAY | DESCRIPTION |
|---|---|---|
| PRTZ | LPRTZ(6) | Contains user directed computational options. If,<br>LPRTZ(1) > 0; Suppress DESIG/Number Print<br>LPRTZ(2) > 0; Suppress TARGET/COMPLEX Print<br>LPRTZ(3) > 0; Ignore grouping<br>LPRTZ(4) > 0; Renormalize target value, nothing else<br>LPRTZ(5) > 0; Suppress FLAG/DESIG Print<br>LPRTZ(6) > 0; Group only |
| SET | RANGEMOD | Fraction of weapon system range for grouping use |
| | RETARGET | Nonzero for missile retargeting capability |
| | CCREB(20) | Command and control reliability for regions |
| | NCCREL | Number of input values for CCREB |
| TARCLAS | MAXCLAS | Maximum number of target classes |
| | INCLAS | Number of user selected target classes |
| | INDESIG(15) | DESIG of exemplar target for selected class |
| | EXPVAL(15) | Value of selected exemplar target |
| WEAPON | MAXW | Maximum number of weapon systems permitted for processing |
| | IWEAP | Number of user selected weapon systems |
| | INWEAP(100) | Names of user selected weapon systems |
| | IREFW(100) | Reference codes of user selected weapon systems |

## 5.7 Subroutine ENTMOD

| | |
|---|---|
| PURPOSE: | Read and store user inputs and control flow of supporting subroutines |
| ENTRY POINTS: | ENTMOD (first subroutine called when overlay link PLANS is executed) |
| FORMAL PARAMETERS: | None |
| COMMON BLOCKS: | CPRIOR, INZ, PRTZ, SET, TARCLA, WEAPON |
| SUBROUTINES CALLED: | CINSGET, GRPEM, INSGET, PRINTW, SRTTGT |
| CALLED BY: | COP |

Method:

In addition to controlling the flow of supporting subroutines, ENTMOD mainly reads and stores user input data (figure 22). The verb and adverbs recognized by this module are:

- PLANSET - the verb that causes execution
- SETTING - the adverb which introduces a clause to set parameters RANGEMOD, RETARGET or CCREL
- PRIORITY - the adverb which introduces a clause to set criteria for choosing representative targets of complexes. Criteria is ordered lists of TASK and alpha-portion of the DESIG
- ATTACKERS - the adverb which introduces a clause to select weapon system inventory. Input is a list of values for attribute TYPE
- DEFENDERS - The adverb which introduces a clause to select target classes. Inputs are a pair of words for DESIG (and hence the target class that the DESIG defines) and exemplar value of each DESIG.
- ONPRINTS - the clause which permits the user to control the computational flow. Since PLANSET is normally executed many times for or. scenario, many of the calculations and prints are redundant. This clause provides a means of suppressing outputs and, hence, increasing throughput time.

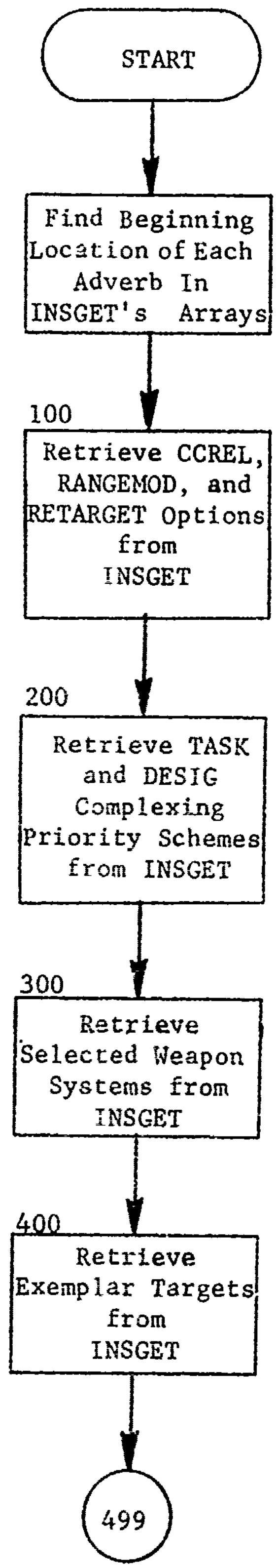


Figure 22. PLANSET Module (Part 1 of 2)

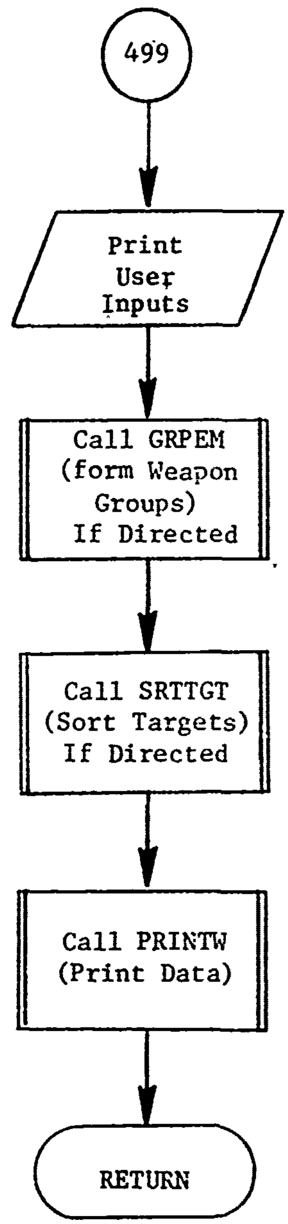


Figure 22. (Part 2 of 2)

## 5.8 Subroutine ADJUSTGP

PURPOSE: Adjust number of bomber refuels so that it is less than or equal to the number of tankers and, further, reset yield for each group

ENTRY POINTS: ADJUSTGP

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30, KEEP, WEAPON

SUBROUTINES CALLED: DIRECT, HDFND, ITLE, MODFY, NEXTTT, RETRV

CALLED BY: GRPEM

Method:

After all data base items have been processed, the total number of bomber refuels (NBOMB) and tankers (NTANK) are compared. If there are more bomber refuels than tankers, the bombers on the nonalert base with the largest range are changed to nonrefuel (IREFUEL=0); NBOMB is decremented by the number of bombers on the base. This process continues until there are more tankers than bomber refuels. If IREFUEL is changed to zero for all the nonalert bases before the bomber-tanker balance is achieved, the alert bases are then examined and IREFUEL is changed as above. When the bombers and tankers have been balanced, the yield for each group is computed. During grouping, individual group records were created but never modified if additional bases were to be added. These additions will be included and the accumulated yields scaled according to the number of weapons within each group.

Subroutine ADJUSTGP is illustrated in figure 23.

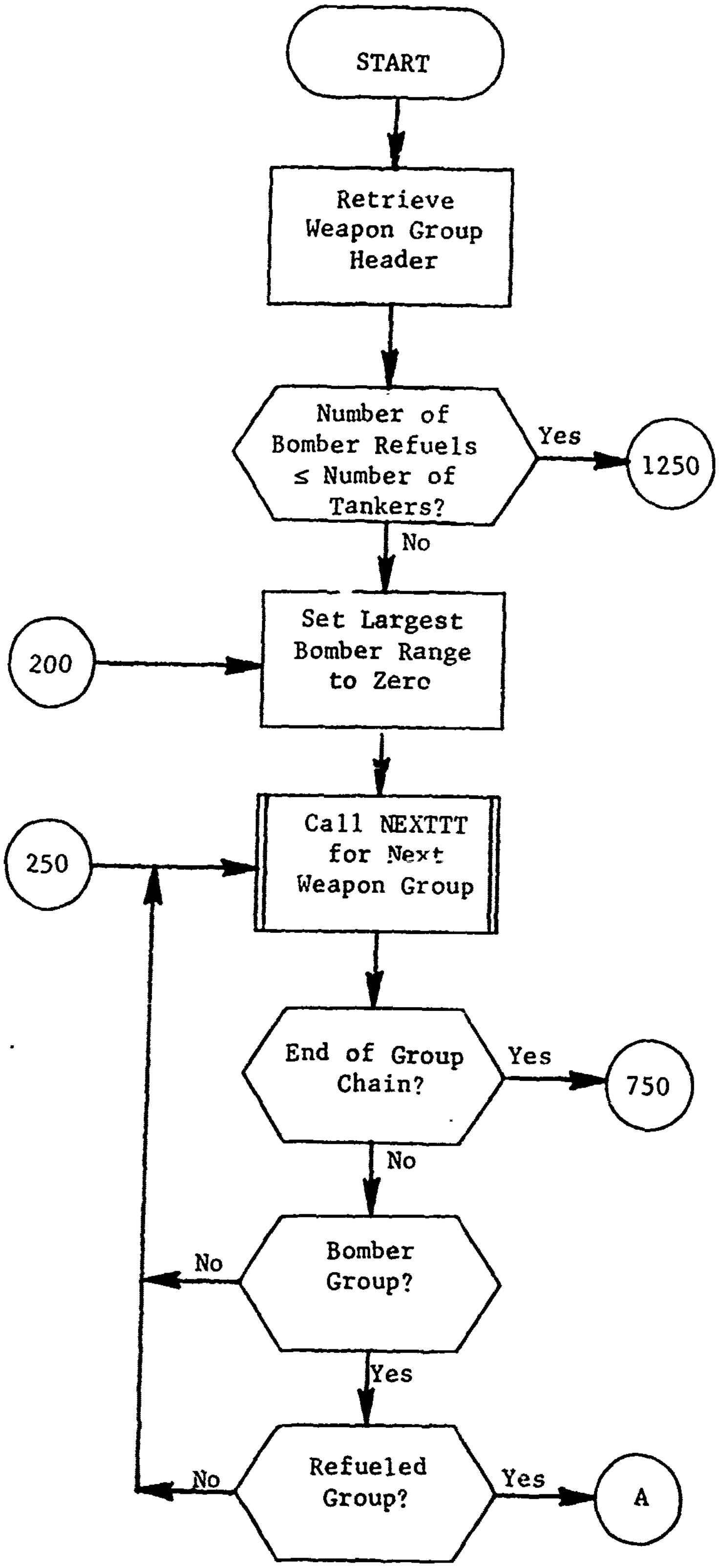


Figure 23. Subroutine ADJUSTGP (Part 1 of 4)

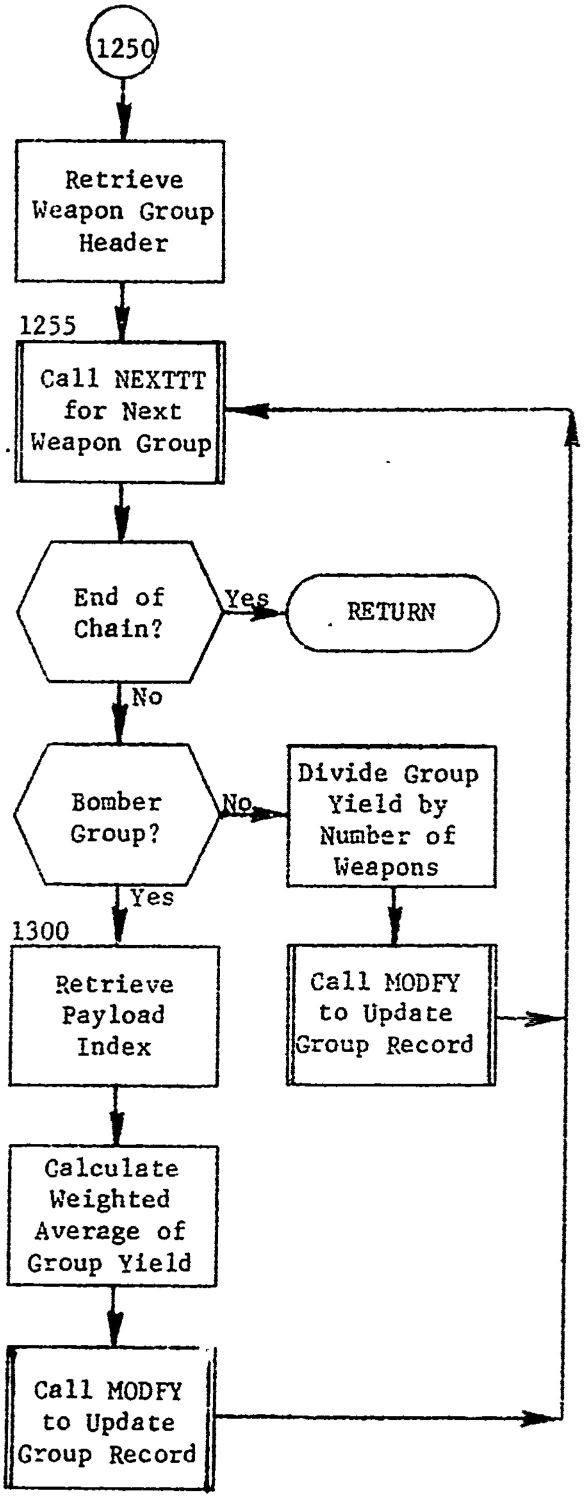


Figure 23. (Part 4 of 4)

## 5.9 Subroutine CALCOMP

| | |
|---|---|
| PURPOSE: | To calculate data which represent a complex target from data for the elements of the complex |
| ENTRY POINTS: | CALCOMP |
| FORMAL PARAMETERS: | None |
| COMMON BLOCKS: | CPRIOR, C10, C15, C30, KEEP, MASK |
| SUBROUTINES CALLED: | DIRECT, HEAD, IGETHOB, MODFY, NEXTTT, ORDER, REORDER, VALTAR, VLRADA |
| CALLED BY: | SRTTGT |

Method:

A search is made for the maximum target radius, which is assigned as the radius of the complex. Similarly, the maximum value of TARDEF is found and assigned to represent the complex. The target value VOZ, the number of terminal interceptors (MISDEF), and the weighted (by VOZ) attributes MINKILL and MAXKILL are accumulated as each target is encountered. Also, for each target, the time components and the corresponding actual value lost at that time are placed sequentially in the arrays V and TAU. The DESIG of each target component is compared against each input of the list of priorities in order to choose the "representative" target for the complex.

Subroutines ORDER and REORDER are called to arrange the elements of TAU in numerical order and to place the elements of V in the corresponding order. Those ordered arrays are used to approximate the time dependence values for the complex (T1, T2, T3, T4, T5) in the following manner.

First the array TAU is checked for equal time components. If any are found, the corresponding values are added together, and all equal components but the last are set to zero. When the entire array has been checked it is collapsed to eliminate any zero components. If the number of remaining entries does not exceed five the time dependence of the value is approximated by these time components. Otherwise, an elimination procedure to reduce the number of entries to five is begun. To accomplish this, the slopes (change in value per change in time) are calculated for all remaining value points and the value point that produces the smallest slope is grouped together with its neighboring value point. Hence the length of the TAU array is reduced by one. The TAU array is repetitively collapsed again, and slopes recalculated until there are five or less points remaining.

Once the elimination process is complete, the fractional value is computed for the first two components from the sums now stored in V(1) through V(5). These fractions, together with the time components in TAU and the total number of components (KK), are stored in array ITAR.

The lethal radius for air bursts must be recalculated for a uniform height of burst for all elements within the complex. This is required since the air lethal radius as calculated from VLRADP (called from PLANSA) assumed an optimal air height of burst for each target. Clearly, one height of burst is required for an air burst over a complex. That height of burst is defined in CALCOMP as the optimal scaled height of burst associated with the hardest element in the complex. The smallest ground lethal radius is defined as being the hardest element in the complex.

Calculation continues to determine the hardness components (HAZ, HGZ, HAZ2, HGZ2) and the corresponding fractional value (FVULN1) which represent the complex. VOZ, FVULN1, and the hardness number (1 or 2) are also recalculated based on the defined scaled height of burst. The complement of FVULN1 is found to represent the second hardness component. If either fractional value is nonzero, it is multiplied by VOZ to get the actual value at that hardness. After all targets have been considered, the lethal radii are separated into radii belonging to hard targets (radii less than 1.5 nautical miles) and radii belonging to soft targets. The average lethal radius, weighted by the actual value at the corresponding hardness, is calculated for both hard and soft targets for those radii. Similarly, the actual value at each hardness (VHARD or VSOFT) is accumulated. If there are no hard targets (i.e., VHARD=0), FVULN1 is set to 1; otherwise the fraction of actual value for hard targets (VHARD/VTOT) is assigned to FVULN1.

After all targets in the complex have been processed as above, the stored values are defined with the target record called 'COMPTG'.

See figure 24 for the logic flow within CALCOMP.

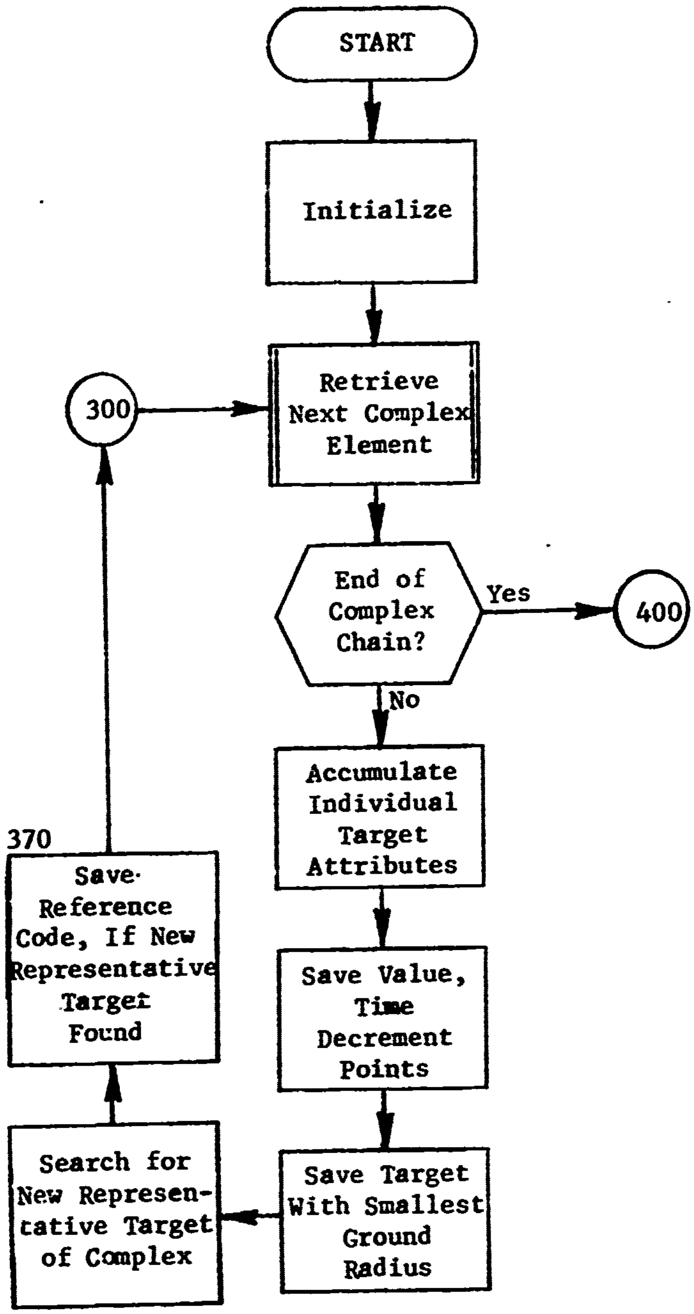


Figure 24. Subroutine CALCOMP (Part 1 of 4)

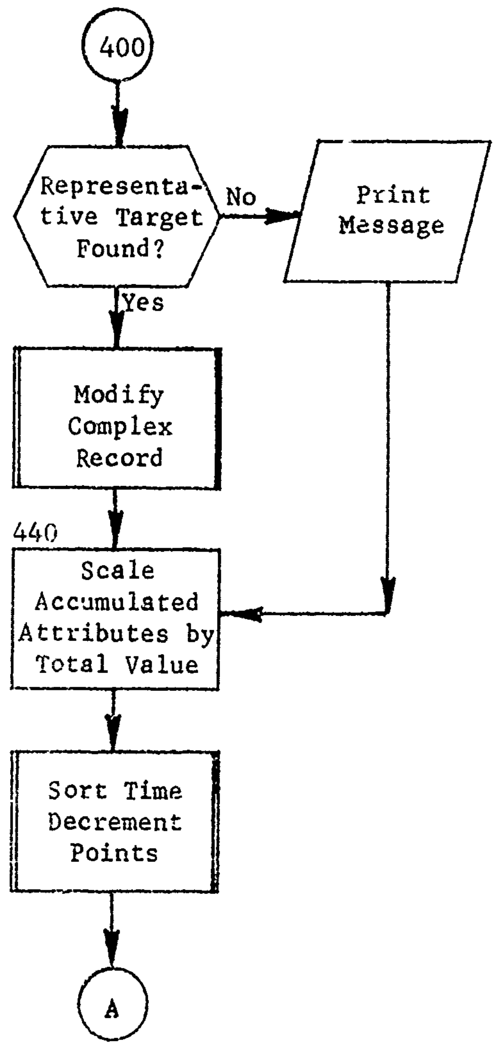


Figure 24. (Pr-t 2 of 4)

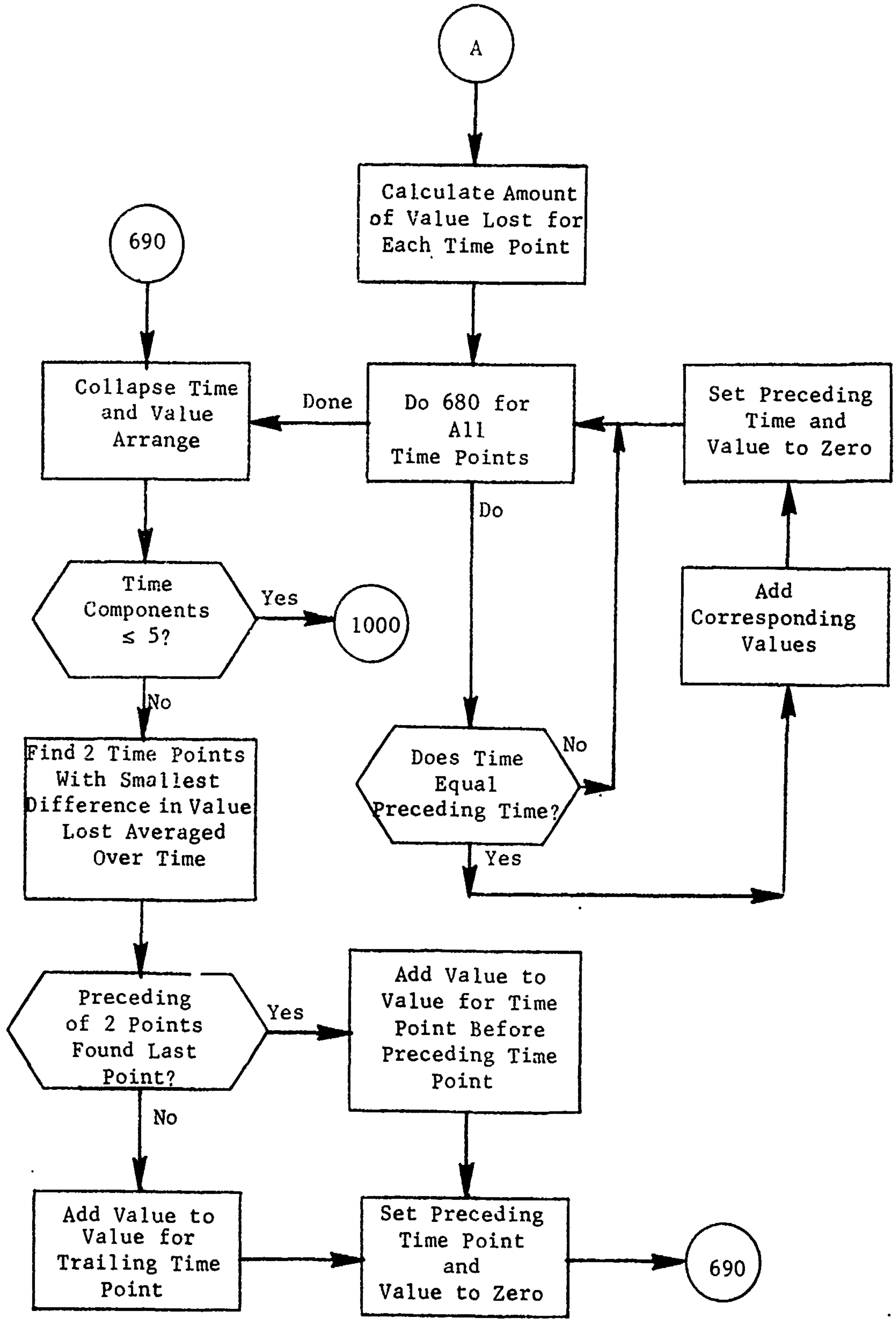


Figure 24. (Part 3 of 4)

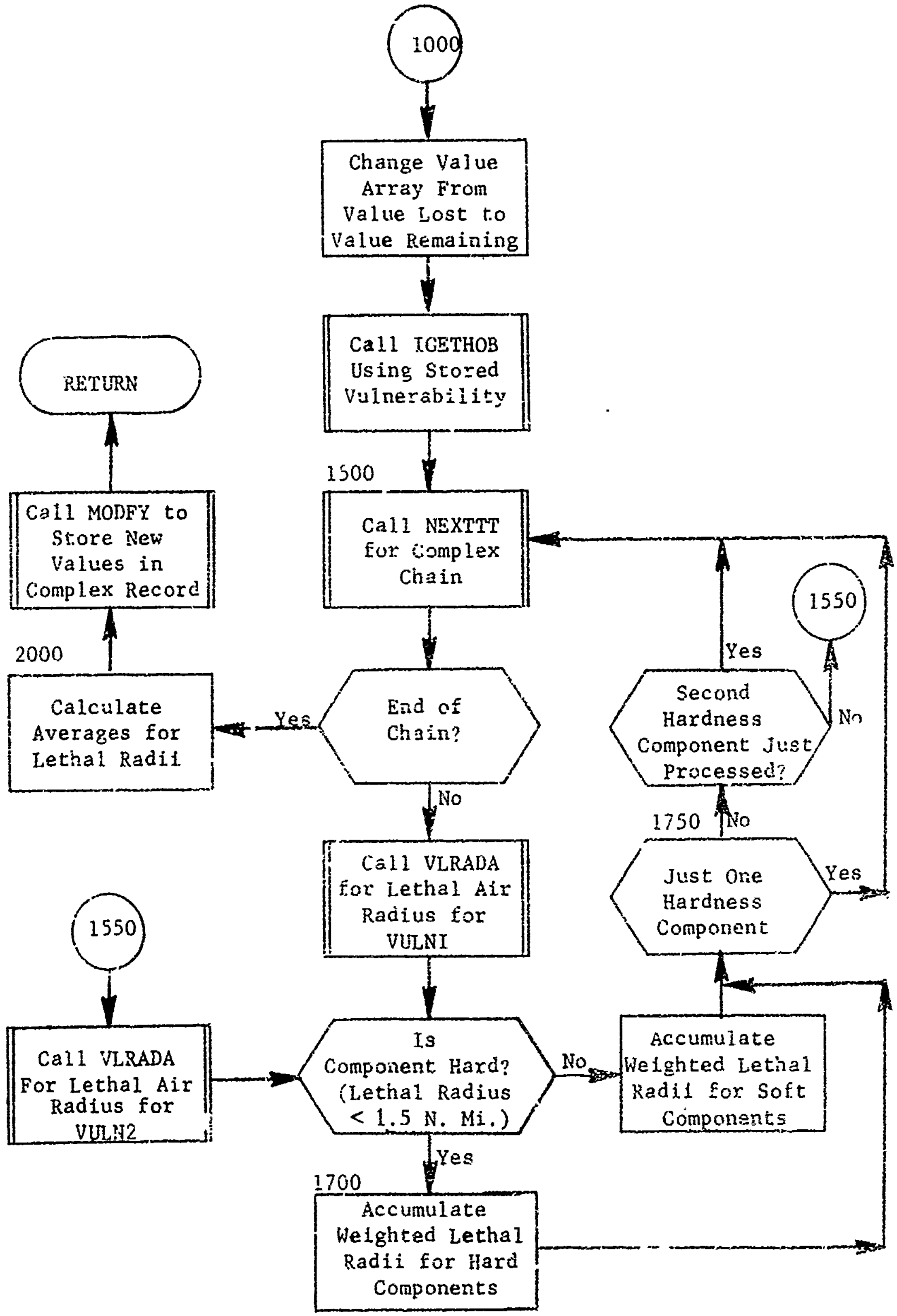


Figure 24. (Part 4 of 4)

THE CONTENTS OF THIS PAGE INTENTIONALLY DELETED

## 5.10 Subroutine GRPEM

PURPOSE: To form weapon groups

ENTRY POINTS: GRPEM

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C20, C25, C30, KEEP, SET, WEAPON

SUBROUTINES CALLED: ADJUST, DIRECT, DLETE, HDFND, HEAD, ITLE, MODFY, NEXTTT, PRINT, STORE, TANKER

CALLED BY: ENTMOD (of overlay link PLANS)

Method:

GRPEM forms weapon groups by chaining the user selected weapon type records ('WEPTYP') and for each type, the individual weapon launch bases are chained ('MSBMTIG'). Each base record then is tested for proper definition under weapon group heading. The grouping order will be performed in the same sequence as the user selected weapon types are.

Tankers are not grouped but are collected and reformatted within subroutines TANKER and ADJUSTGP.

In the case of a missile weapon system, GRPEM checks the retargeting flag (RETARGET). If on, the user has requested that the data base attribute IREP be considered for all missiles. GRPEM then calculates and stores for the current missile type, the factors that later will be used to modify the number per squadron, number on alert, alert DBL probability, and reliability for all missiles of the type.

After weapon type data has been selected and defined, missiles and bombers are aggregated to form weapon groups. A weapon group consists of weapons from up to 150 bases. If all the weapons on a given base are nonalert, weapons of the same type are considered as one group. Otherwise, a group comprises those weapons on a base which have the same alert status, type (attribute TYPE), region, and payload. Bombers must also have the same refueling index. The maximum number of warheads allowed per group is set at 1,000. Also, for missile classes the maximum number of weapons per salvo is set at 15; if exceeded, a new missile group is formed.

Only those records are processed that define the first site of a squadron (ISITE positive).

BOMBER units which do not refuel and missile sites must lie within a geographic region which, for alert weapons, has a radius equal to a certain percentage of the range of the weapon. This percentage is read into the variable RANGEMOD at the beginning of the program; if the percentage is not specified in the data cards, it is assumed to be 15%. For nonalert weapons, the distance criterion is automatically doubled.

In order to form a weapon group, the required radius is expressed in terms of latitude (DLAT) and longitude (DLONG), and the number of bases (NTOTBAS) is counted. If some bombers are to be used as tankers for refueling purposes (i.e., if IREFUEL=2), the number in commission and the number on alert are cut in half. The number of weapons and total yield of the warheads carried by each vehicle on the base then are computed. Up to 250 groups can be formed for use in plan generation. However, PLANSET processes and prints information for up to 260 weapon groups to enable planners to adjust their data base should more than 250 groups be formed.

When a new group is started group data are retrieved and stored under record 'WEPNGP'. For each weapon launch base, the base record ('MSBMTG') is modified and linked to the group header. As each new base is added, the group centroid is adjusted accordingly. If there are both alert and nonalert bombers on a given base, the alert bombers are tested for group assignment first using the distance criterion RANGEMOD; the nonalert bombers then are tested using the criterion 2 x RANGEMOD.

Subroutine GRPEM is illustrated in figure 25.

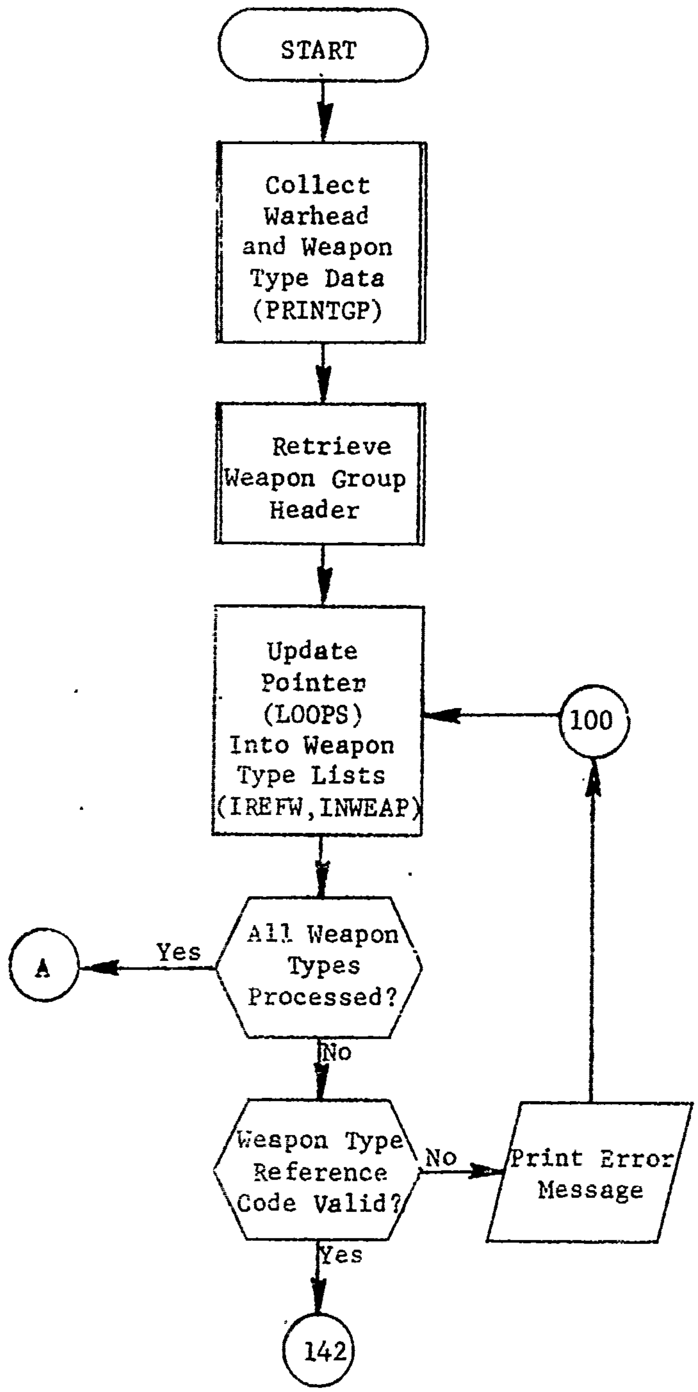


Figure 25. Subroutine GRPEM (Part 1 of 9)

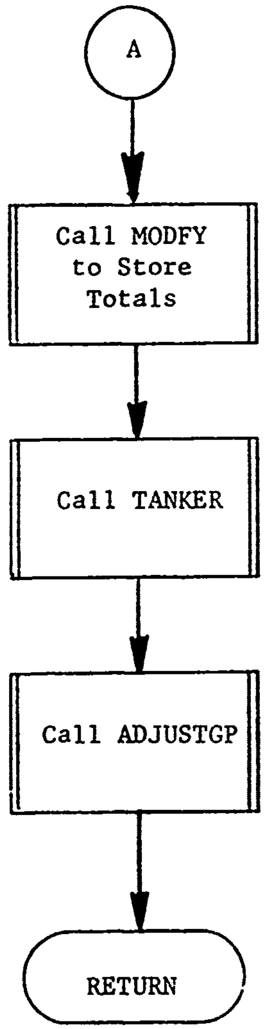


Figure 25. (Part 2 of 9)

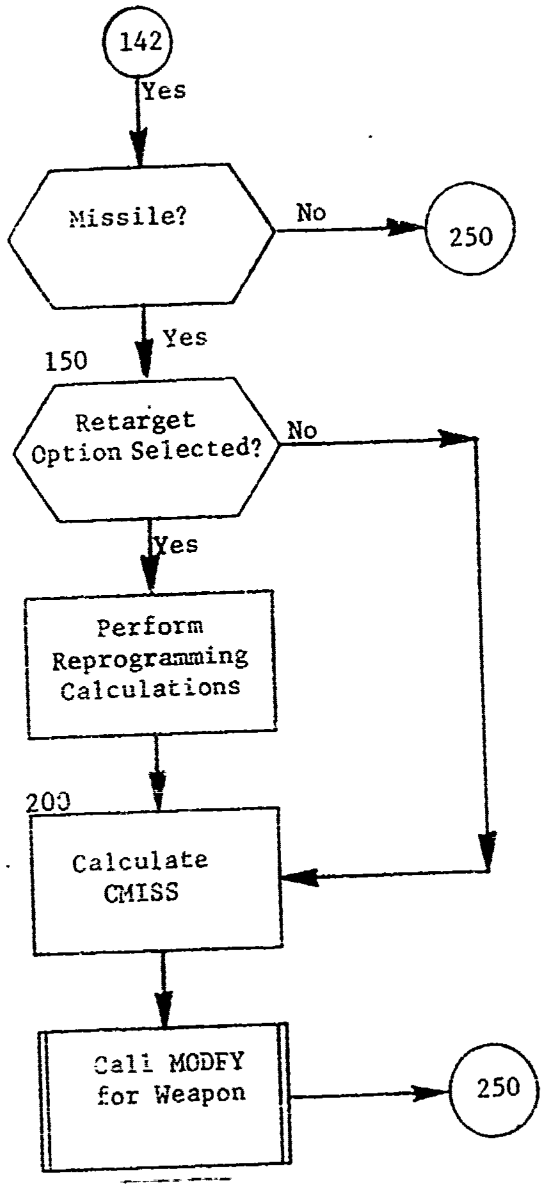


Figure 25. (Part 3 of 9)

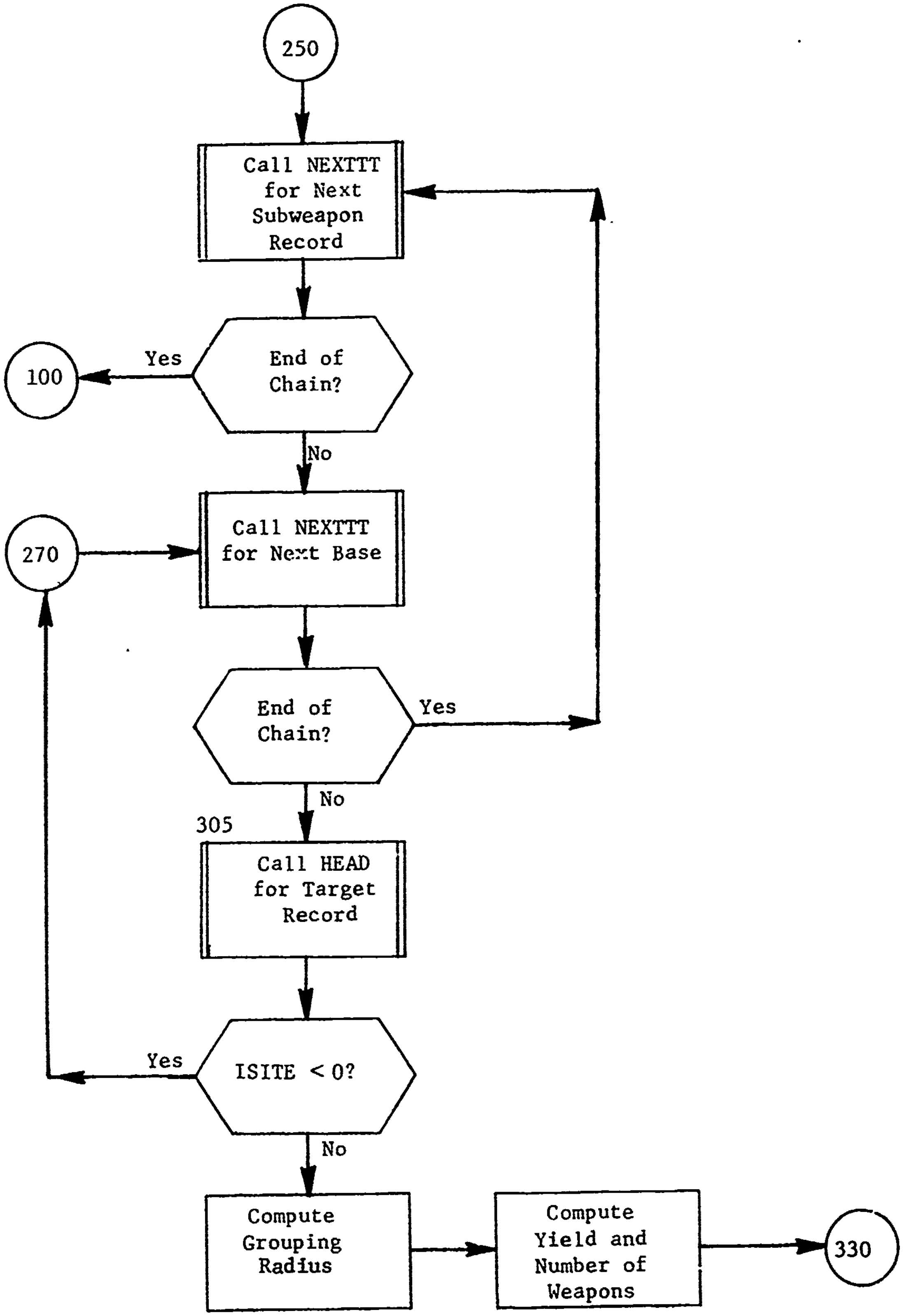


Figure 25. (Part 4 of 9)

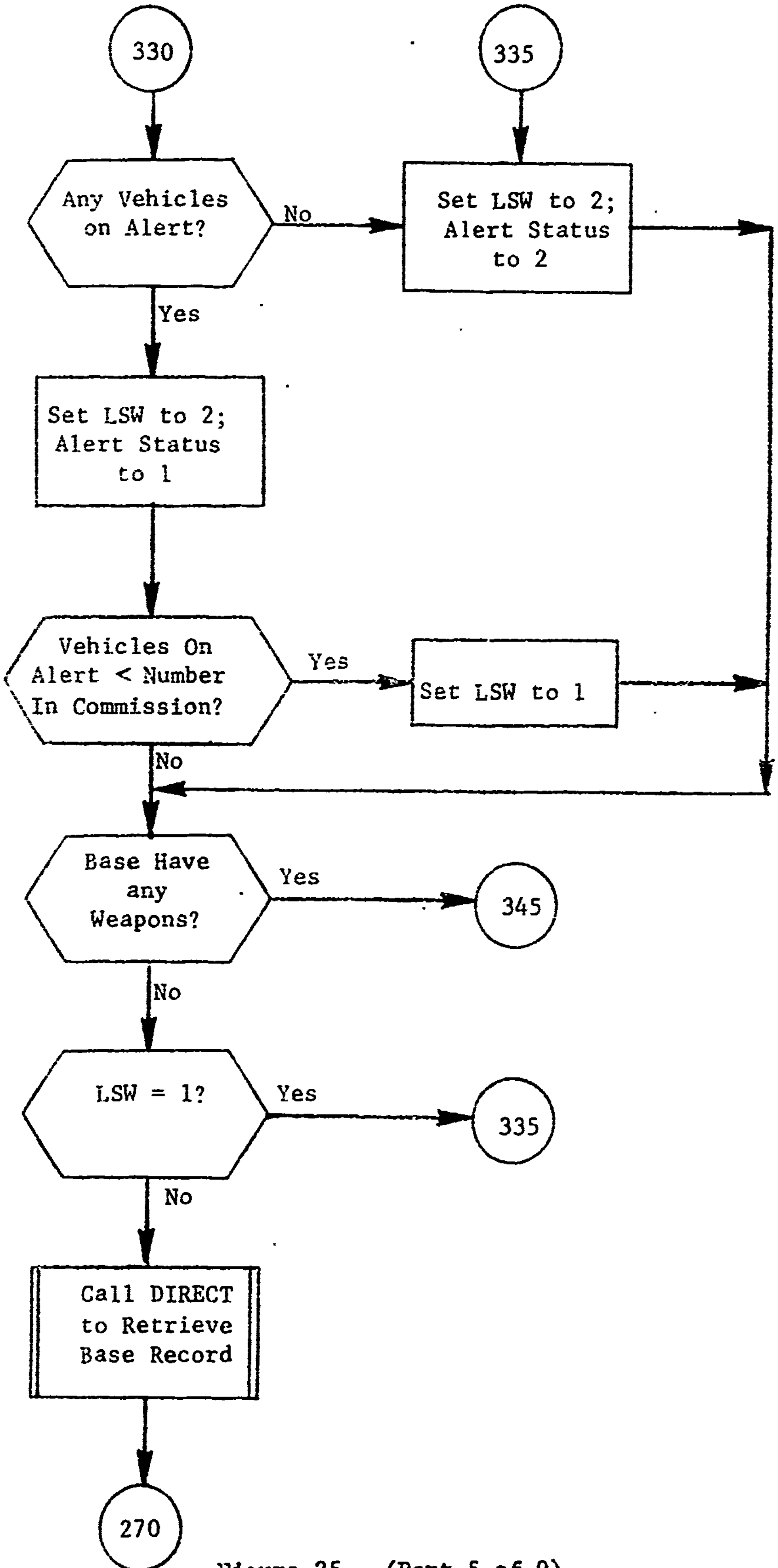


Figure 25. (Part 5 of 9)

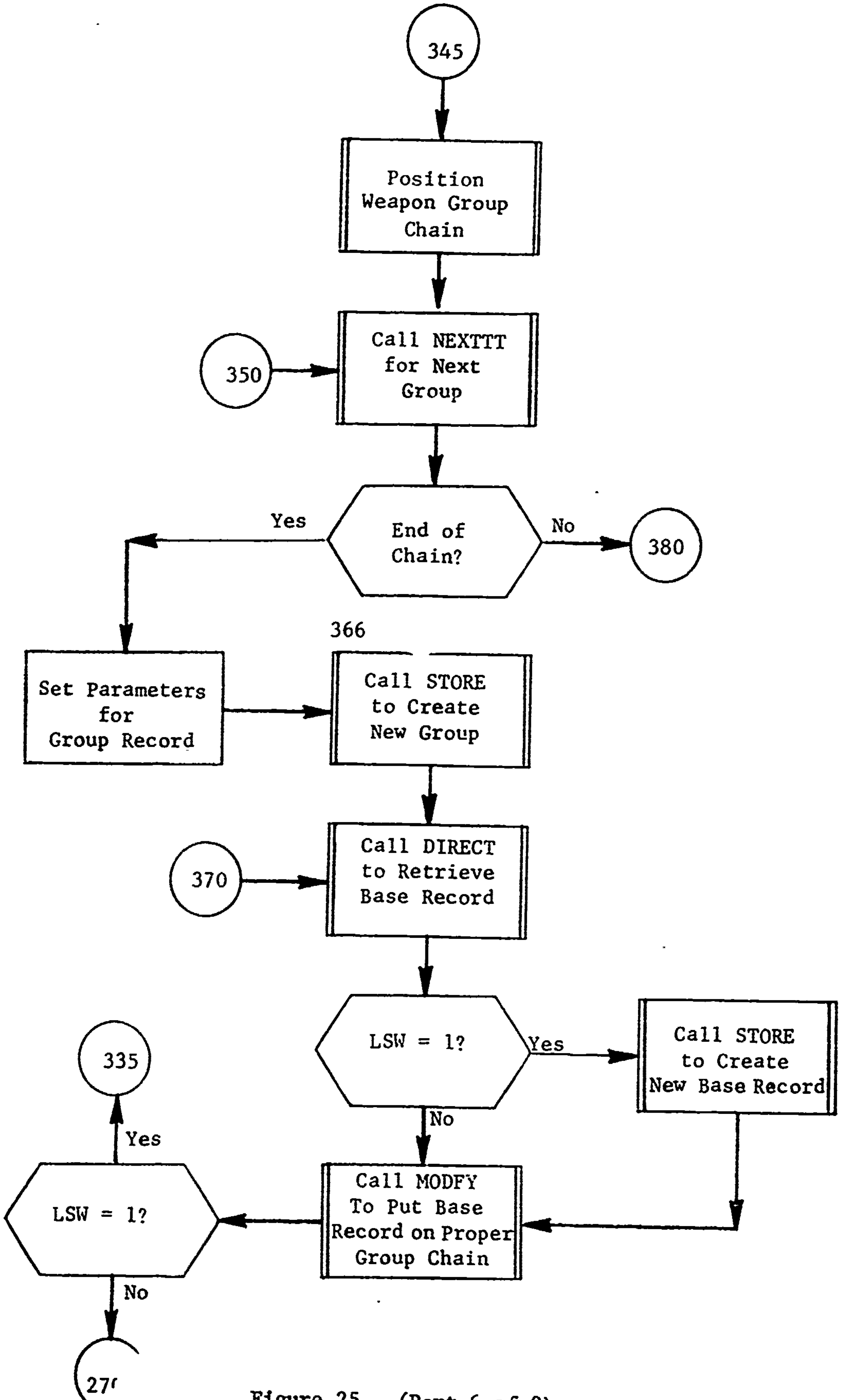


Figure 25. (Part 6 of 9)

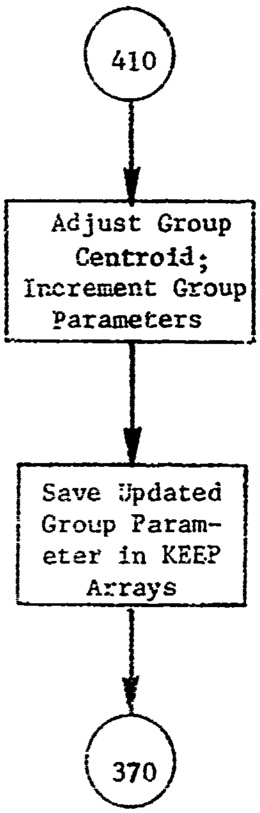


Figure 25. (Part 9 of 9)

## 5.11 Subroutine PRINTGP

**PURPOSE:** Print standard tables

**ENTRY POINTS:** PRINTGP, PRINTW

**FORMAL PARAMETERS:** None

**COMMON BLOCKS:** C10, C15, C30, EXCLAS, KEEP, PBLOK, PRTZ, WEAPON

**SUBROUTINES CALLED:** DIRECT, HDFND, HEAD, ITLE, NEXTTT, RETRV

**CALLED BY:** ENTMOD (of overlay link PLANS)

**Method:**

This subroutine is called twice: once prior to weapon group and target: processing (Entry PRINTGP) and once after all processing has been completed (Entry PRINTW). Execution through Entry PRINTGP permits the collection and print of the Warhead, ASM, Payload, and Weapon Type Characteristics Tables. Weapon grouping requires payload parameters. Therefore, during the collection process these parameters in addition to being printed will be saved (in /KEEP/ arrays) for use within subroutine GRPEM. After processing, Entry PRINTW causes the printing of weapon groups and target, complex lists.

Subroutine PRINTGP is illustrated in figure 26.

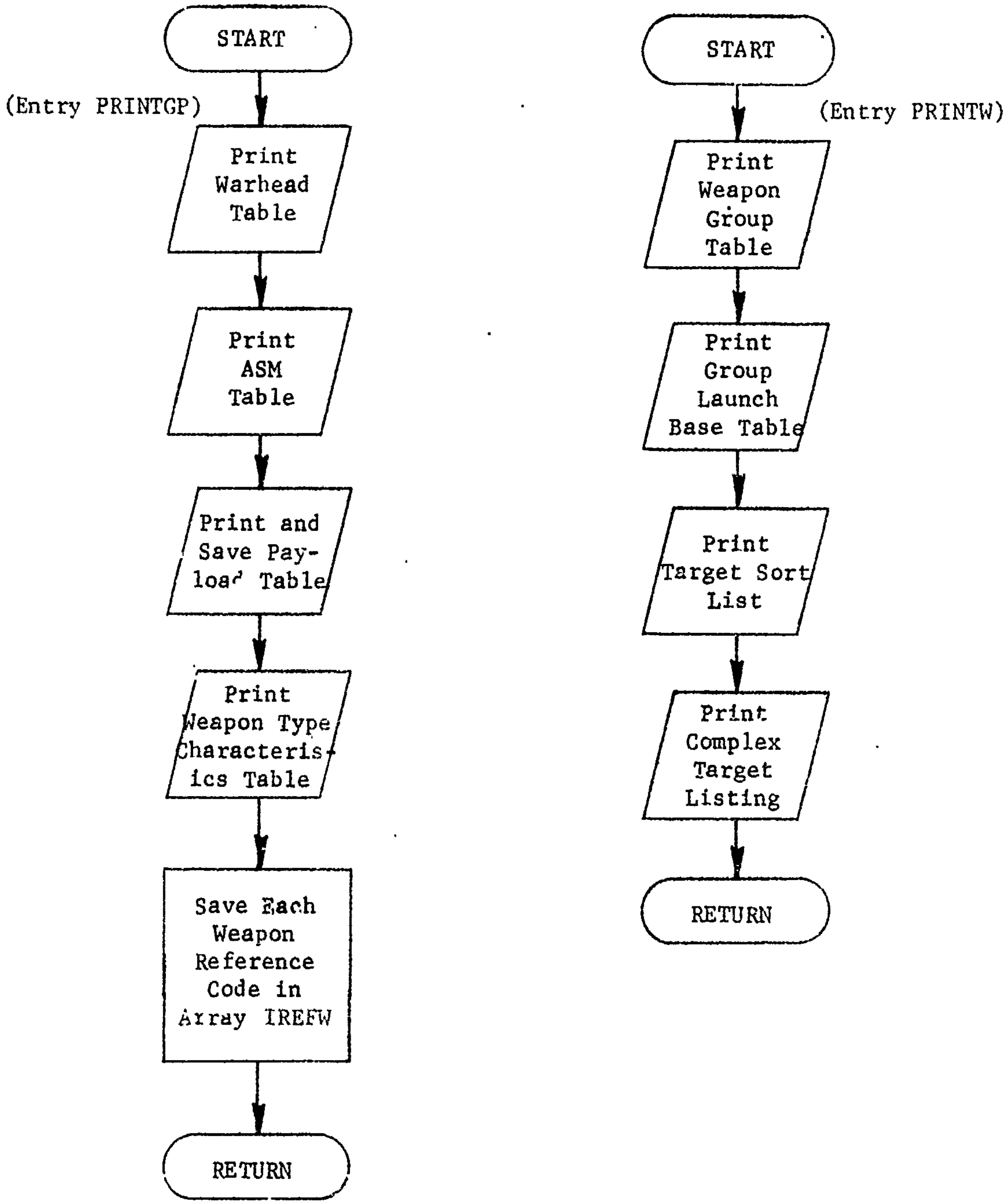


Figure 26. Subroutine PRINTGP

## 5.12 Subroutine SRTTGT

PURPOSE: To process user selected targets

ENTRY POINTS: SRTTGT

FORMAL PARAMETERS: None

COMMON BLOCKS: ERRCOM, CPRIOR, C10, C15, C20, C25, C30, EXCLAS, KEEP, MASK, OOPS, PBLOK, PRTZ, TARCLAS

SUBROUTINES CALLED: CALCOMP, DIRECT, DLETE, HDFND, HEAD, IGET, IPUT, ITLE, KEYMAKE, MODFY, NEXTTT, ORDER, RETRV, STORE, VLRADP

CALLED BY: ENTMOD (of overlay PLANS)

Method:

SRTTGT begins by counting the number of characters in the task inputs (used for choosing representative targets of complexes in CALCOMP) and setting ISUBT to zero or one if the task inputs are one or two characters long, respectively.

Next the exemplar target DESIGs and corresponding values are processed. For each target class to be considered by the allocator, a DESIG and a corresponding value are entered. The DESIG pertains to a target within the target class and the value is its target value before the sum of target values are normalized to 1000. All targets within that class will have their data base values adjusted by a similar ratio before normalization. The exemplar targets are retrieved directly on the CALC chain.

Two passes are made through the entire target list. During the first pass target values are accumulated for each target class, lethal radius stored for individual targets, and counts of individual targets and number of elements in each complex collected. The number of elements in each complex is stored in array NELEM indexed by attribute ICOMPL. The second pass over the target list will use these counts. Parameters related to optional prints are also stored during this pass. Attribute DESIG is temporarily stored onto scratch file ITEMP if the Target DESIG/Target Number print was selected. If directed (Print option six equals zero), attributes FLAG and DESIG are packed into one word stored in working array KEY1, and eventually sorted using a generalized sort/merge beginning at label 9000.

Before the second pass, scaling factors which will adjust relative target values and normalize the sum of target values are computed. Also, parameters for assigning randomized target numbers are computed.

After the first pass over the target list is completed, elements of complexes not selected by the user (through the exemplar target technique) are transferred from the complex chains they reside on to the simple target chain. When all such targets are transferred, each complex that has one element remaining is also transferred to the single target chain.

The second pass over the target list assigns the final target value, the randomized target number, and forms representative elements of complexes by calling subroutine CALCOMP.

Upon processing each target record, the target number can be immediately calculated and stored. The calculation is: a sorting index (LEAD), which is a function of the total number of targets (NTAR), is determined by the formula:

$$\text{LEAD} = \frac{\text{NTAR } (3 - \sqrt{5})}{2}$$

To start a cycle, a beginning index (IBEG) is designated and assigned to the target being read. Initially IBEG=LEAD. The index for the next target (IND) then if found by incrementing the previous index (LAST) by LEAD. If the result exceeds NTAR, the cycle is reset by subtracting NTAR from IND. When a cycle is completed (i.e., when IND=IBEG) the next cycle is begun by incrementing IBEG by one and proceeding as above. Thus, a unique nonsequential index (TGTNUMB) is assigned to each target as it is read. TGTNUMB along with the target reference code (TGTREFCD) is temporarily written onto scratch file IF4 for eventual sort and final storage.

Each complex must be processed (by CALCOMP) once <u>after</u> each element in a given complex has its target value assigned. The first pass updated array NELEM (later transferred to KEY1) to define the number of elements in each complex. The second pass updates a second array (KEYC) in the same fashion. By comparing the counts between KEY1 and KEYC, CALCOMP may be executed at the appropriate time.

After the second pass, optional prints are provided by reading files IR and ITEMP. Also, file IF4 is read, sorted on TGTNUMB and the TARCDE records created.

Current data base sizes demand that all interfaces with IDS be kept at a minimum. To accomplish this, common block KEEP is constantly being redefined as working storage arrays that holds pertinent data. Since sizing is open-ended, arrays contained in /KEEP/ must provide for spilling its contents onto scratch files in the desired sort order. Logic beginning at label 9000 initiates code that performs a basic sort/merge process. Working arrays, are sorted via the standard ORDER subroutine and merge with any part information stored on scratch files. This sort/merge process is open-ended and processing is swift.

Subroutine SRTTGT is illustrated in figure 27.

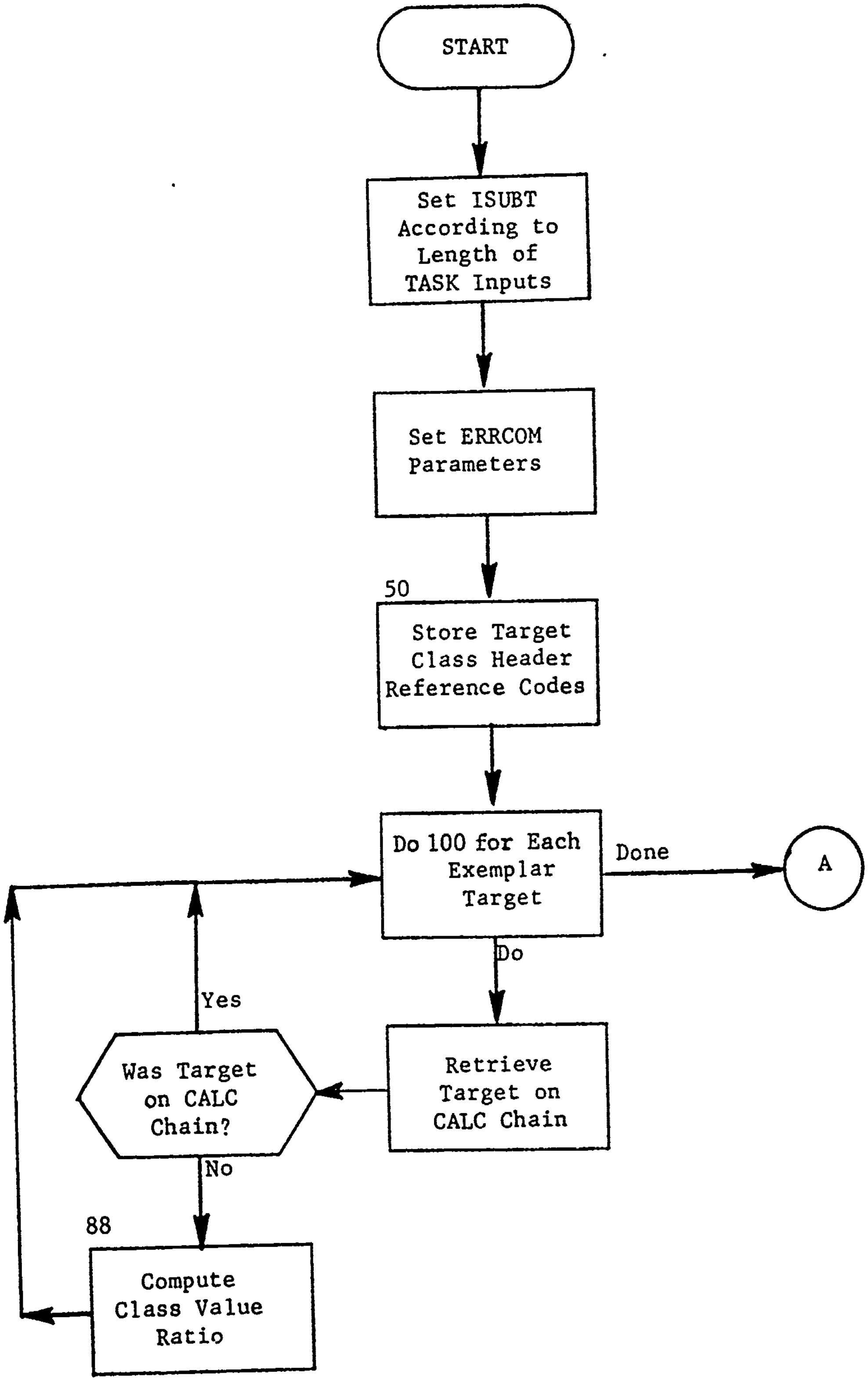


Figure 27. Subroutine SRTTGT (Part 1 of 11)

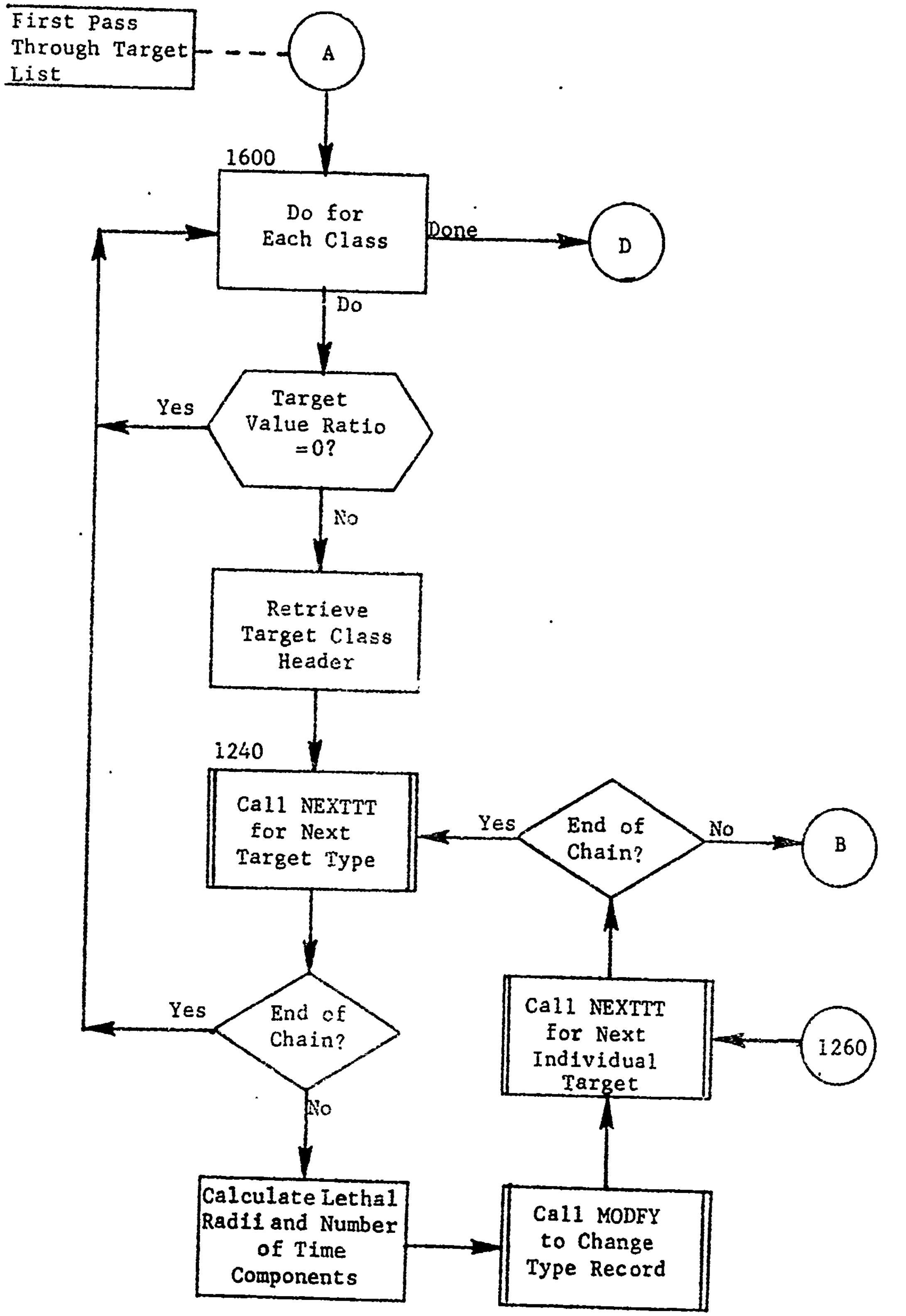


Figure 27. (Part 2 of 11)

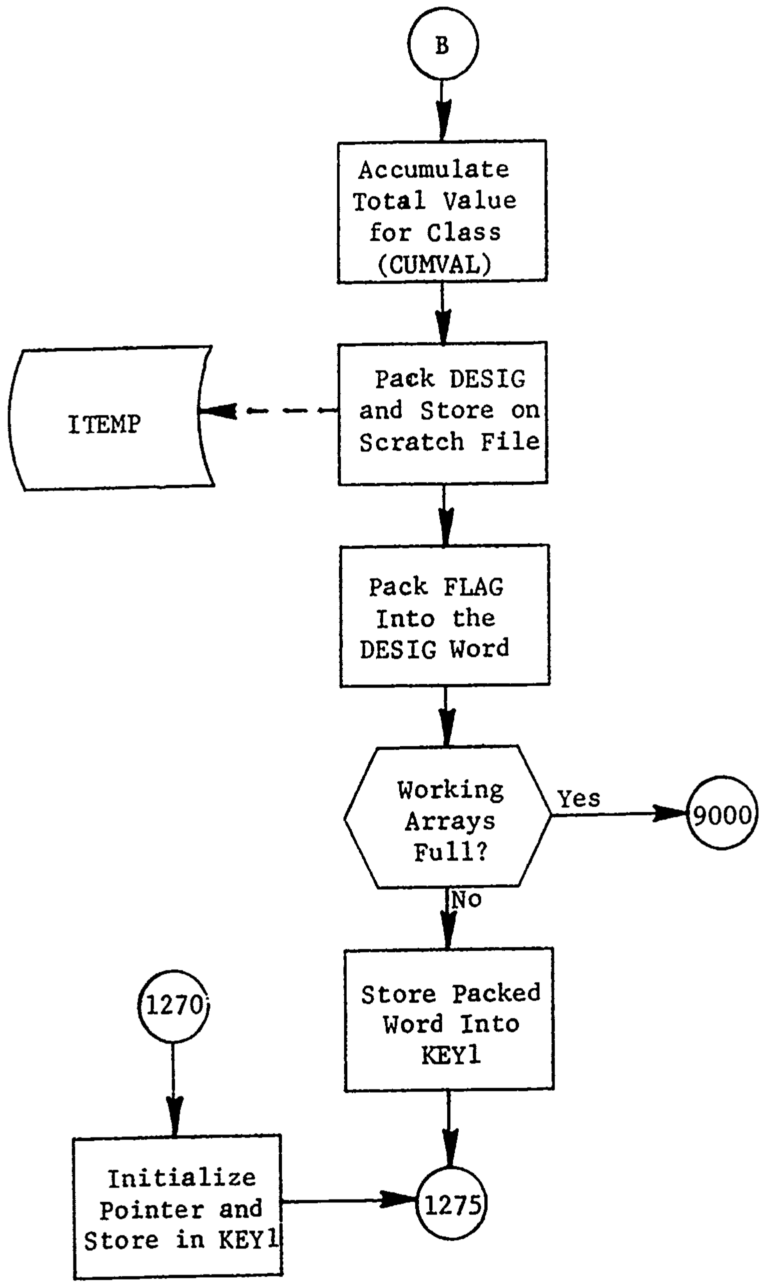


Figure 27. (Part 3 of 11)

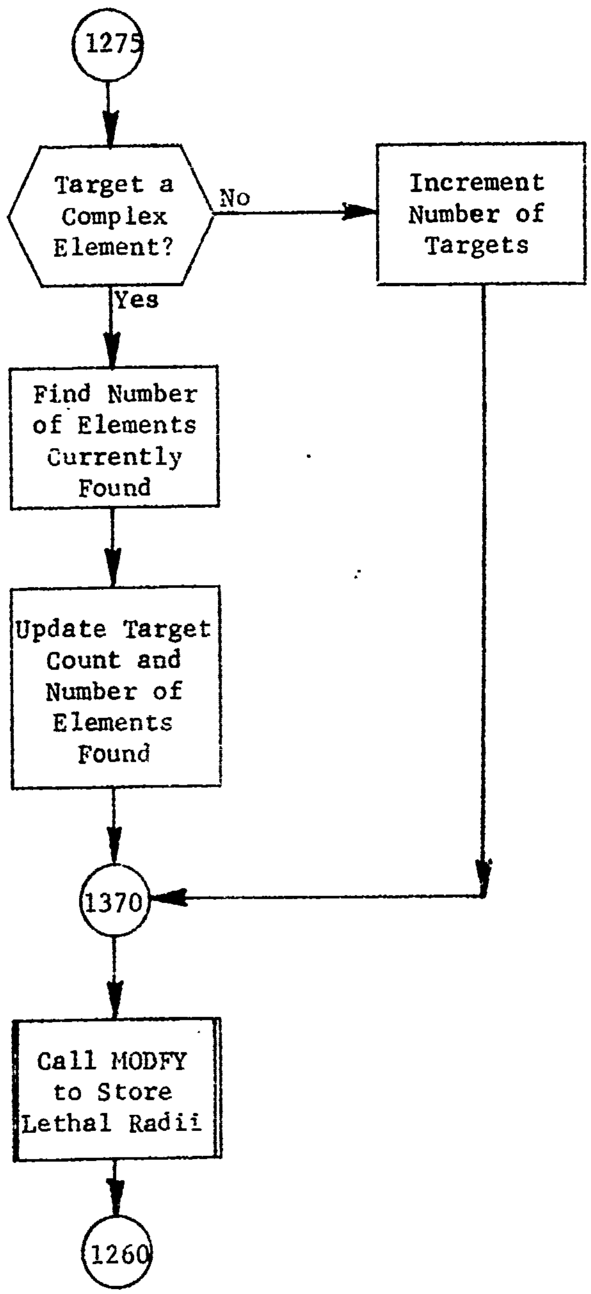


Figure 27. (Part 4 of 11)

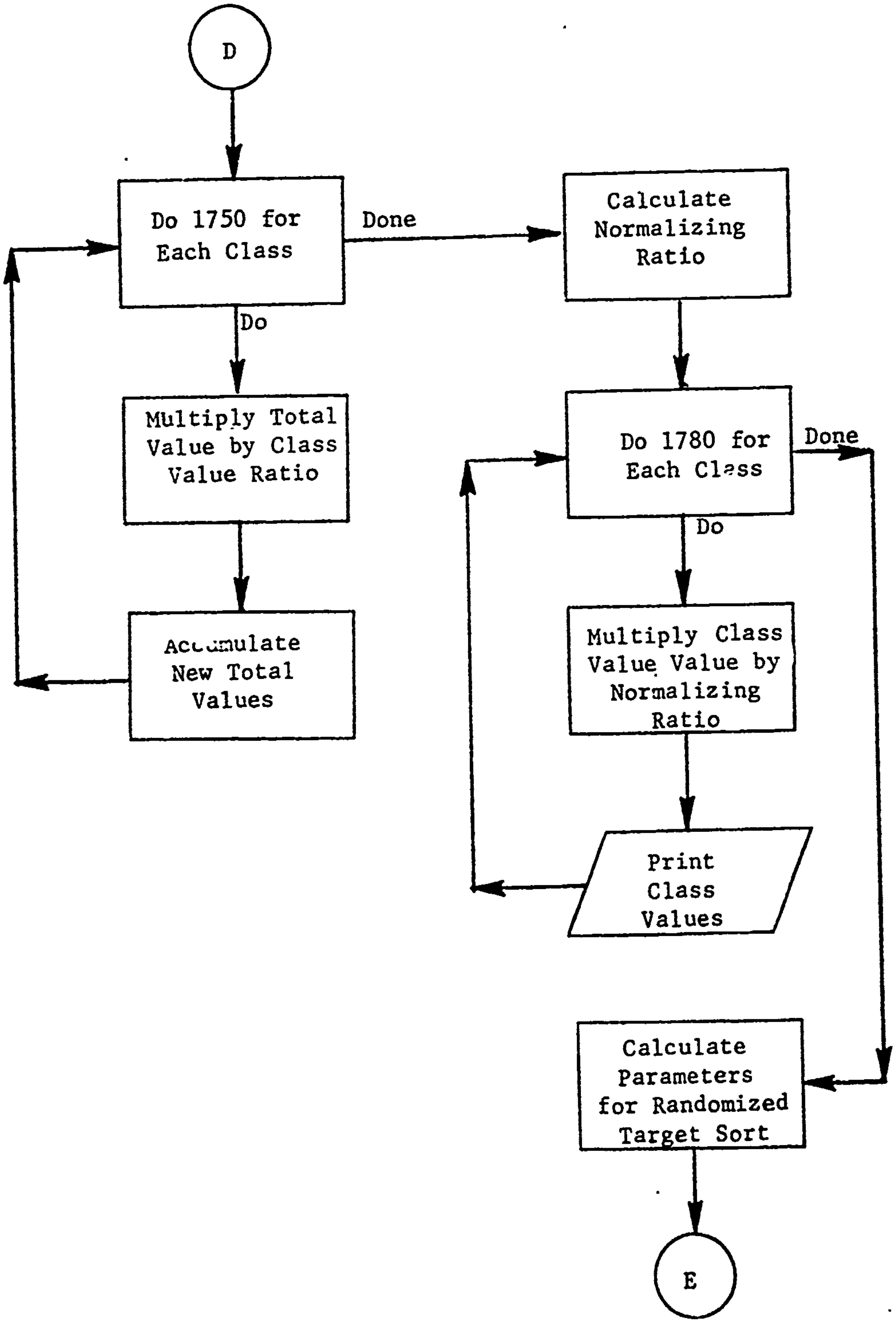


Figure 27. (Part 5 of 11)

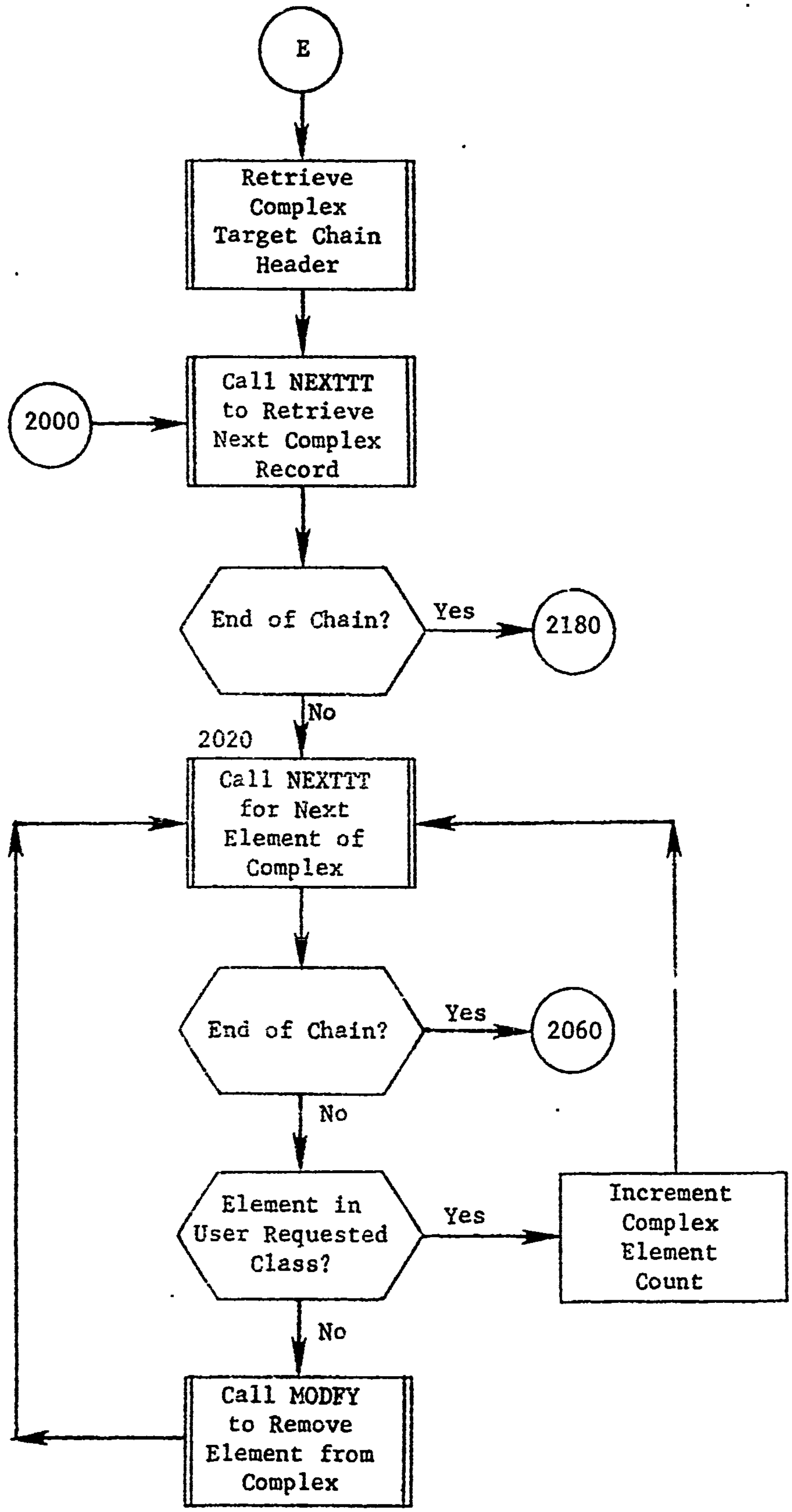


Figure 27. (Part 6 of 11)

THE CONTENTS OF THIS PAGE INTENTIONALLY DELETED

THE CONTENTS OF THIS PAGE INTENTIONALLY DELETED

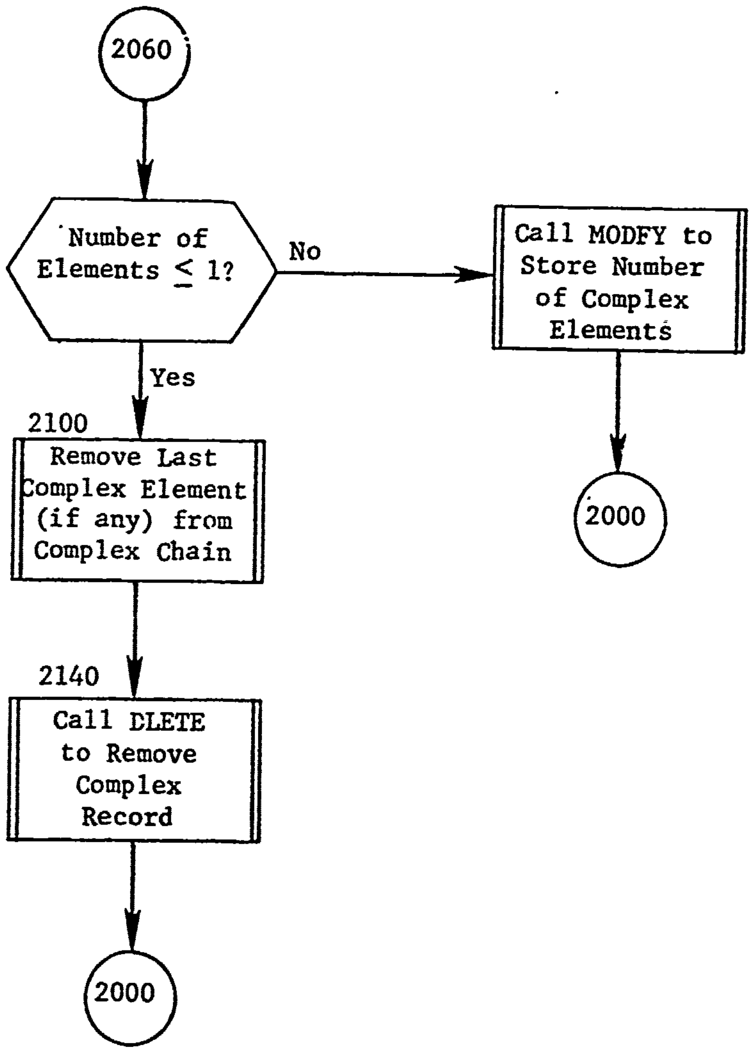


Figure 27. (Part 7 of 11)

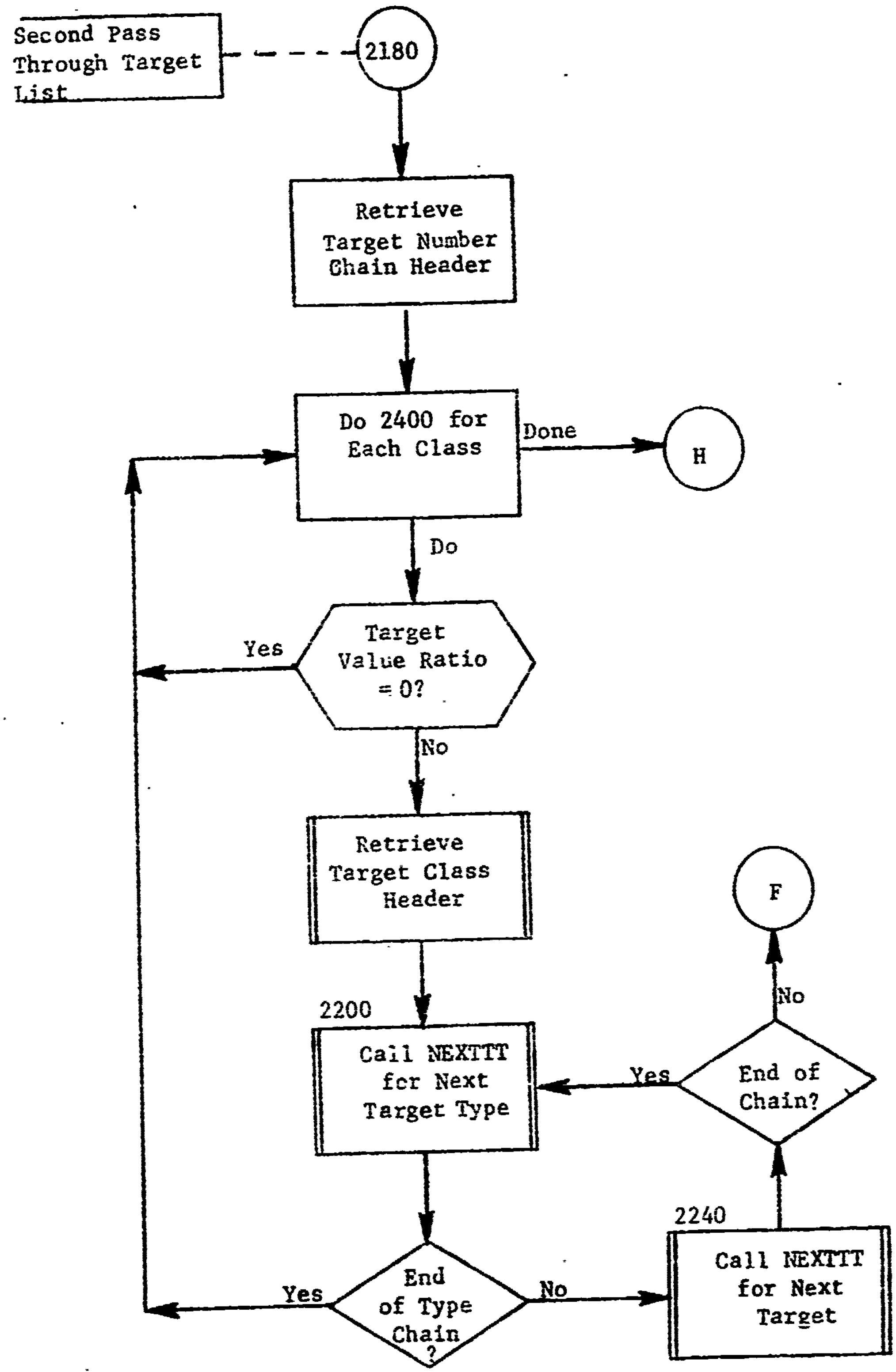


Figure 27. (Part 8 of 11)

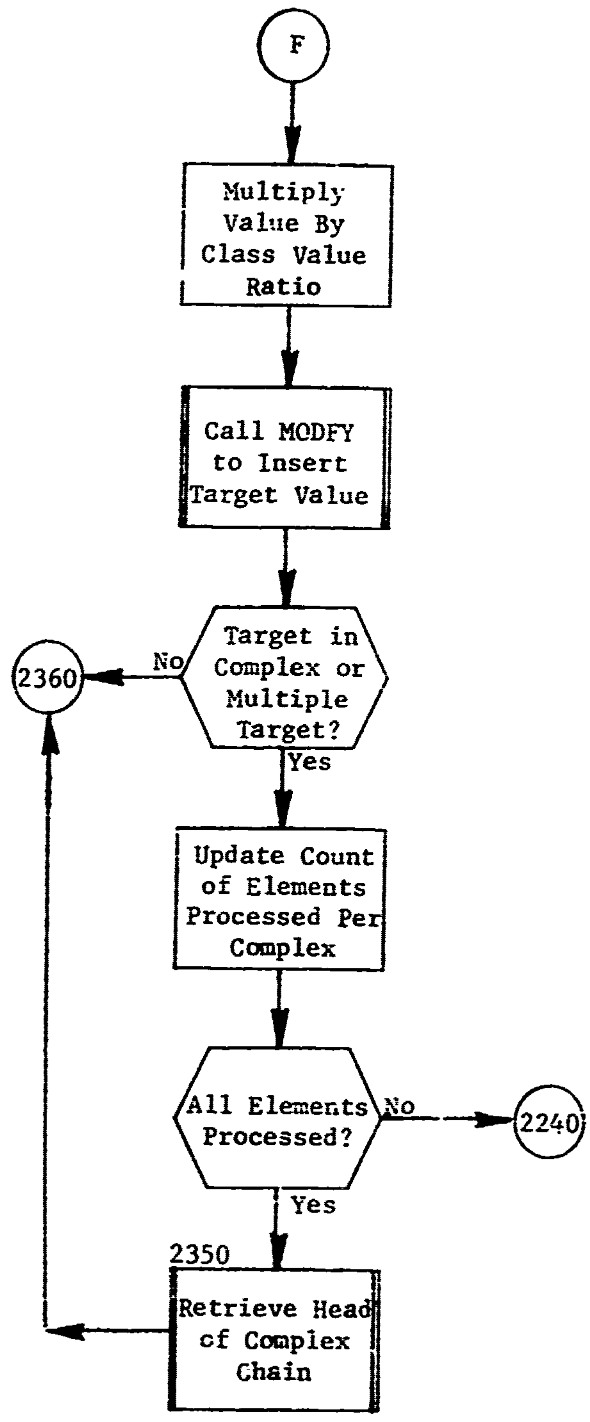


Figure 27. (Part 9 of 11)

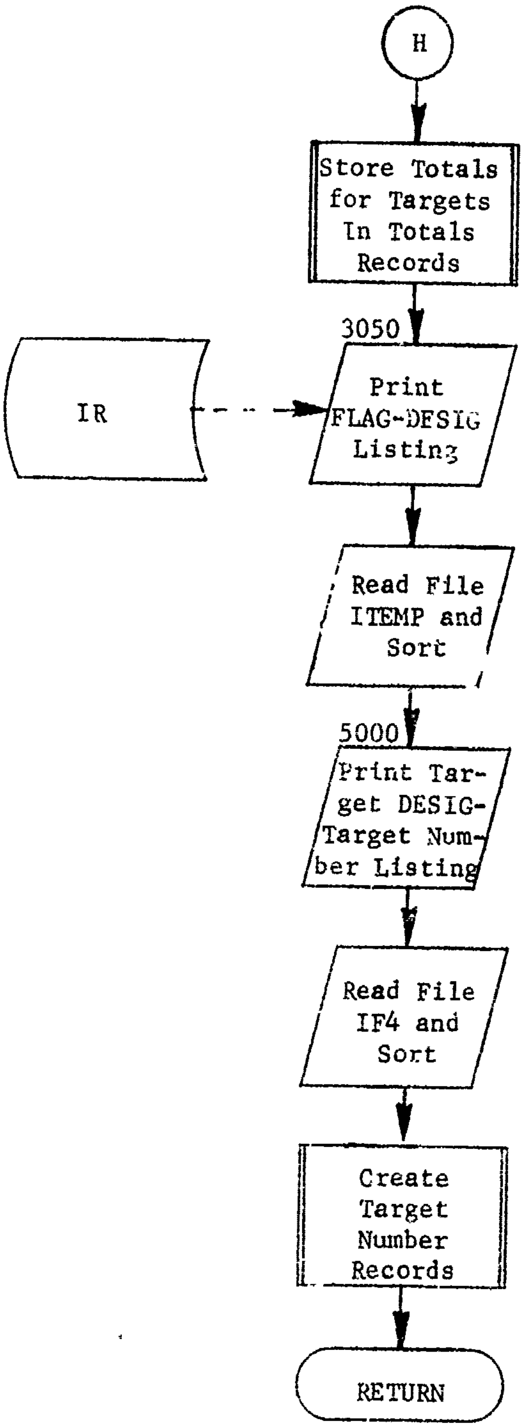


Figure 27. (Part 10 of 11)

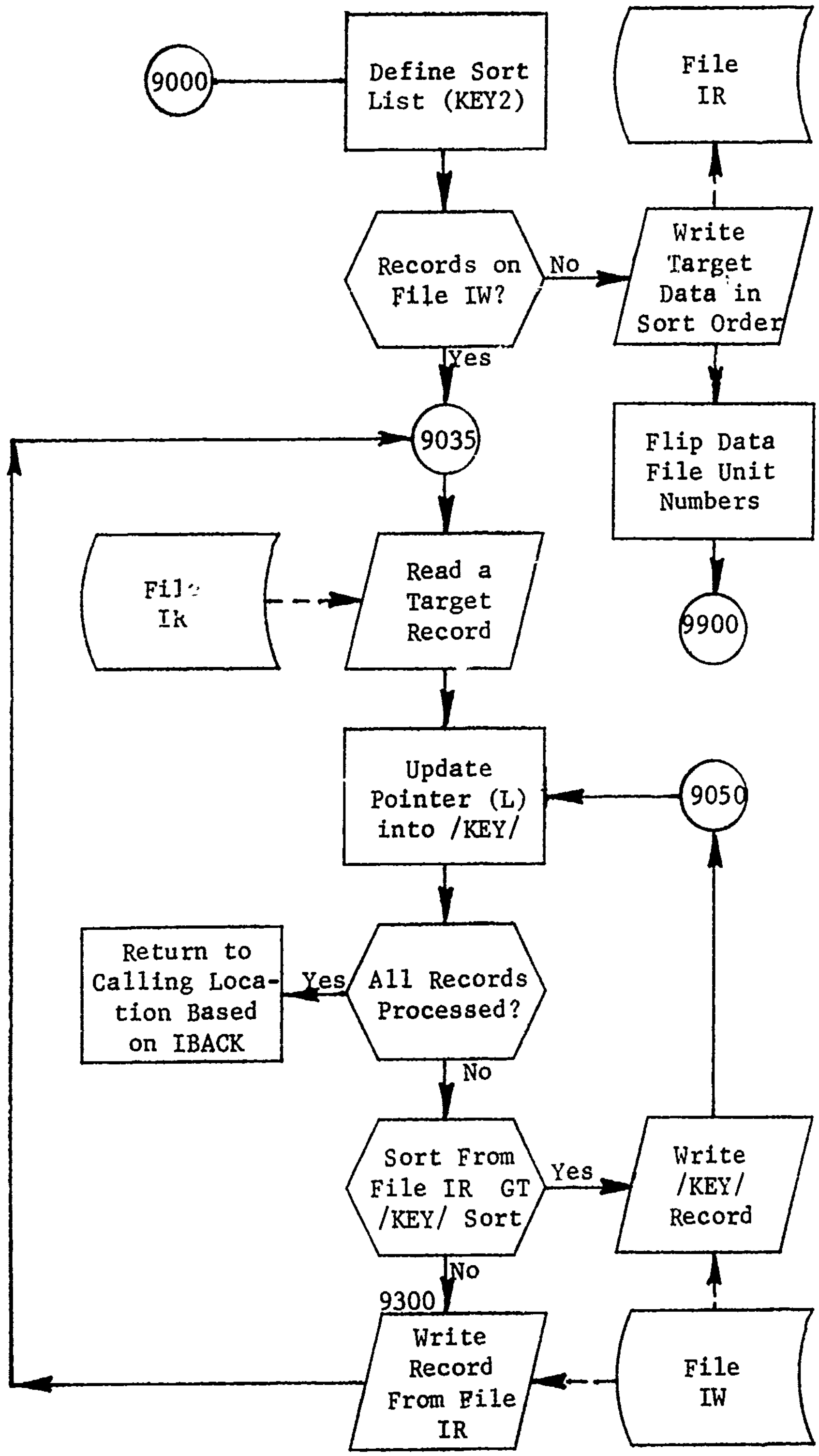


Figure 27. (Part 11 of 11)

Then, the basic yield, YIELD(G), for a group is calculated as

$$\text{YIELD(G)} = \left[\frac{\text{NOBOMB1} * \text{YIELD1}^{(2/3)} + \text{NOBOMB2} * \text{YIELD2}^{(2/3)}}{\text{NOBOMB1} + \text{NOBOMB2}}\right]^{3/2}$$

It is this basic yield which is used for all bombs of the group during the allocation phase (module ALOC). The actual yield of the ASMs is used for these weapons.

MRV/MIRV Payloads: In QUICK, those missiles equipped with a multiple reentry vehicle (MRV) capability are allocated to a single target. For allocation purposes, the component RVs (reentry vehicles) are considered to be a single warhead; however, the added effect of the MRV's pattern is reflected in the formula used to determine its expected yield:

$$\text{MRV yield} = (\text{yield for one warhead of the given type}) * (\text{the number of warheads, or RVs})^{3/2}$$

The number of warheads (reentry vehicles) is raised to the 3/2 power in order to accommodate the "2/3 rule" for comparing the yield of N MRV warheads delivering X megatons each against the yield of one warhead of NX megatons striking the target center.

Multiple independently targetable reentry vehicles (MIRVs), on the other hand, are allocated as separate weapons, subject to footprinting constraints. Hence, for the case in which the independently targetable reentry vehicles (IRVs) of a missile with MIRV capability are in turn equipped with MRVs, the expected yield calculated is:

$$\text{Yield for missile with MIRV capability} = (\text{yield for one warhead of the given type}) * (\text{the number of IRV's}) * (\text{the number of warheads, or RV's, per IRV})^{3/2}$$

Command and Control Reliability: Each weapon item in the data base is assigned to a command and control region (IREG) by the user. This command and control region is an arbitrary designation for the extent of command and control functions and has no geographic meaning. The reliability for command and control (CC) is a function of this region IREG. Thus, the user must divide the offensive weapons systems into these "regions" according to the command and control which is appropriate for the plan being developed. The maximum number of command and control regions is 20.

Overallocation: The QUICK weapon allocator is designed to assign the individual weapon of a group to specific targets. In developing this allocation, program ALOC does not consider serial bombing constraints or MIRV footprint constraints. These constraints reflect the physical limitations on a delivery vehicle's ability to deliver warheads to geographically separated targets. In addition, in allocating bomber weapons, the number of weapons associated with a given penetration corridor may not correspond to an integral number of delivery vehicles.

The above constraints are considered in the sortie generation phase of plan development. To provide some flexibility in developing feasible weapon assignments for each delivery vehicle, a few extra weapons are added to each MIRV and bomber weapon group for allocation by module ALOC. Subsequent processing by modules POSTALOC (for bombers) and FOOTPRNT (for MIRVs) removes this overallocation in creating the sortie specifications.

For this reason, a number of weapons are artificially added to each weapon group. The formula used to add these weapons is as follows:

$$NEX = NWOLD * (PEX + EXB/NVOLD)$$

where NEX = number of weapons added to group
NWOLD = original number of weapons in group
NVOLD = original number of vehicles in group
PEX = percentage extra factor
EXB = extra vehicle factor

There is one set of increase factors (PEX and EXB) each for bombers, non-MIRV missiles, and MIRV missiles. These increase factors are user-input parameters.

This excess of weapons appears as an over-allocation of weapons from the weapon allocation phase. The sortie generation phase removes this over-allocation in creating the sorties. Thus the final number of weapons for which plans are generated closely approximates the number requested in the data base. (In some extreme cases, some weapons may be omitted.)

In order that the Plan Generator will perceive the correct number of expected weapons, the survival before launch probability (SBL) is modified to reflect this change.

If: NACTUAL = actual number of weapons in a group
NEXCESS = number of weapons added to the group

then: $$SBL = SBLREAL * \left[\frac{NACTUAL}{NACTUAL + NEXCESS}\right]$$

The actual survival before launch probability (SBLREAL) is used after the excess weapons have been removed in the sortie generation phase of plan development.